91802
new 82-10

LEGAL NOTICE

This report was prepared by Ontario Hydro and, while effort has been made to assure the accuracy and applicability of the material contained herein, neither Ontario Hydro nor any person acting on its behalf:

(a) makes any warranty, express or implied with respect to the use of any information, apparatus, method or process disclosed in this report or that such use may not infringe privately owned rights; or

(b) assumes any liabilities with respect to the use of, or for damages resulting from the use of, any information, apparatus, method or process disclosed in this report.

COMPLIMENTARY COPY
Please Credit Ontario Hydro Research if Reproduced in Whole or in Part

ontario hydro
research division

HYDRAULIC COMPRESSION OF A SHAFT BACKFILL - ANALYTICAL MODEL DEVELOPMENT

Report No 90-347-K

K.C. Lau, Engineer
H.T. Chan, Senior Engineer
Soil Sciences Section
Civil Research Department

ABSTRACT

A one-dimensional computer code, based on the Integrated Finite Difference technique, has been developed to describe the deformation (compression and expansion) behaviour of a clay-based backfill under a large hydraulic load. The analytical model has been validated with one set of compression test results on a partially saturated, compacted backfill specimen consisting of 25% slightly swelling Lake Agassiz clay from the Winnipeg area and 75% granular material. A reasonable comparison between the laboratory data and the analytical results was obtained.

43-110-061	843.2	March 27, 1991	90-347-K

TABLE OF CONTENTS

	Page
1.0 INTRODUCTION	1
2.0 OBJECTIVES OF THE RESEARCH	2
3.0 DEFORMATION OF SOIL	2
3.1 Principle of Effective Stress in Saturated Nonswelling Soils	3
3.2 Principle of Effective Stress in Partially Saturated Nonswelling Soils	4
3.3 Principle of Effective Stress in Saturated Swelling Soils	6
3.4 Principle of Effective Stress for Partially Saturated Swelling Soils	7
4.0 DEFORMATION OF SOILS UNDER A HYDRAULIC LOAD	7
4.1 Saturated Nonswelling Soils	8
4.2 Saturated Swelling Soils	8
4.3 Partially Saturated, Nonswelling Soils	9
4.4 Partially Saturated Swelling Soils	9
5.0 THE PROPOSED ANALYTICAL MODEL	10
5.1 Analytical Model	11
5.2 Calibration of the Analytical Model with Preliminary Laboratory Data	14
5.2.1 The One-Dimensional Compression Test	14
5.2.2 The Analytical Prediction	15
5.2.3 Comparison of Results	15
6.0 CONCLUSIONS	16
REFERENCES	18
FIGURES 1 TO 10	19
APPENDIX A - Computer Code Listing	29
APPENDIX B - Material Properties	61
DISTRIBUTION	last page

ontario hydro
research division

To Mr. D.W. Souther
Manager
Nuclear Engineering Department
Design and Development Division - Generation

Attn: Dr. S.J. Naqvi

HYDRAULIC COMPRESSION OF A SHAFT BACKFILL - ANALYTICAL MODEL DEVELOPMENT

1.0 INTRODUCTION

At the close-out stage of the proposed Canadian underground repository for used nuclear fuel, the dewatering system for the repository vault will be decommissioned. The groundwater regime around the vault will redistribute and return to the equilibrium position. At great depths, the magnitude of the groundwater pressure on the shaft backfill at the backfill-rock interface may cause compression of the shaft backfill. As a result, an annular gap may develop between the shaft backfill and the shaft wall and thereby form a short circuit path for the migration of radionuclides from the vault to the biosphere. The width of this gap and the rate of its closing are functions of the hydraulic load imposed by redistribution of the groundwater, the compressive and swelling behaviour of the backfill mixture, and the size and geometry of the shaft backfill. The backfill mixture should be designed to allow healing of this gap prior to any deterioration of the used nuclear fuel canisters.

Natural occurring geological materials such as swelling clays are considered to have the greatest potential to meet the functional requirements of backfill over long periods of time (thousands of years) under the changing vault environment.

The large size of the shaft and the potentially large hydraulic load involved make it extremely difficult and costly to perform full-scale field tests to assess the performance of various candidate clay-based backfill mixtures. The use of laboratory scale model tests to assess the full-scale performance of a clay-based shaft backfill is limited by problems associated with the inherent scale effect. However, in this study, a one-dimensional analytical model calibrated with the experimental results obtained from the high-pressure compression test cell (maximum pressure 7 MPa) constructed at Ontario Hydro/1/, has been developed

to determine the compressive behaviour of clay-based shaft backfills subjected to large hydraulic loads. This model will be used to assist in the selection of a reference clay-based shaft backfill mixture. In addition, the model can be extended to a three-dimensional model to be used to predict the performance of in situ shaft backfill tests to be conducted at the Underground Research Laboratory in Manitoba.

2.0 OBJECTIVES OF THE RESEARCH

The time-dependent deformation behaviour of a clay-based shaft backfill subjected to a large hydraulic load is a function of the size of the shaft. During the initial compression of the shaft backfill, the amount of diametric compression of the shaft backfill is proportional to the diameter of the shaft. With the ingress of water into the shaft backfill, the backfill will start to expand as a result of the elastic rebound of the backfill, caused by a decrease in effective stress, and the physiochemical swelling of the backfill mixture. The initial expansion of the shaft backfill, a local process occurring primarily at the backfill-rock interface where there is direct contact with groundwater, is essentially independent of the diameter of the shaft. This local behaviour restricts the use of laboratory-scale model test results to assess the performance of the field prototype shaft sealing system. Hence, in this study, an analytical model describing the time-dependent deformation behaviour of clay-based backfills subjected to large hydraulic load is formulated. The analytical model will be used to assess and predict the performance of various candidate clay-based backfill mixtures under full scale conditions.

In this report, the mechanisms that govern the deformation of soils are discussed. The responses of various types of soil at different degrees of saturation subjected to hydraulic loads are examined. The development of the analytical model is described and theoretic results are then compared with the laboratory experimental data obtained in 1990.

3.0 DEFORMATION OF SOIL

The response of a soil element to changes in external load is deformation. The deformation of a soil element on a macroscopic scale is mainly the result of the relative movement of grains, such as sliding, rolling and slipping, and the deformation of the individual soil grains.

In saturated soil element, the mechanism of deformation is mainly of two types: compression and shear. The compression mechanism involves significant volume changes with some or no distortion of the soil element; on the other hand the shearing process involves significant distortion together with some volume

changes of the soil. In actual soils, the mechanism of deformation is neither pure compression nor pure shear. Due to the random arrangement of soil particles, the compression mechanism involves some local distortions and similarly the shear mechanism may involve different degrees of volume changes. However, from an engineer's point of view, the deformations can be evaluated by the technique of superposition of the two mechanisms.

In a partially saturated soil the mechanism of deformation is more complex. The state of stress at the particle contacts (intergranular stress) is governed by the applied stress as well as suction pressure. The deformation of a partially saturated soil element is due to more than one mechanism:

(a) deformation of soil grains,
(b) deformation of the soil skeleton, and
(c) change in the size and shape of the air-water menisci.

While the first two mechanisms are similar to those in saturated soils, the third mechanism is much different from the other two and is not found in saturated soils. When the soil particles slide and roll over one another, the contacts fail, causing air-water menisci to rupture. Although the new contacts are made quickly, the air-water menisci do not readily form again. This results in a change in capillary stresses and also in the intergranular stresses at the particle contacts. Thus, in both of the compression and shearing processes, the changes in suction pressures play an important role in the stress-deformation behaviour of partially saturated soils.

In the next few sections, the stresses that govern the deformation behaviour of soils at different degrees of saturation are presented.

3.1 Principle of Effective Stress in Saturated Nonswelling Soils

A soil is a three-phase material that consists of the soil skeletons, water, and air. For a saturated nonswelling soil, the deformation of the soil is governed by Terzaghi's effective stress concept. This concept is regarded as an axiom of soil mechanics. In the form first used by Terzaghi in his theory of consolidation and subsequently shear strength/2/, the effective normal stress on any plane in soil mass was given by:

$$\sigma' = (\sigma - u) \qquad (1)$$

where σ' = effective normal stress on a plane,
σ = applied normal stress on that plane, and
u = pressure in the pore fluid.

With the determination in the changes in effective stress, the deformation of a soil element can be calculated.

$$\Delta V/V = -m_v * \Delta\sigma' \quad (2)$$

where m_v = the compressibility of the soil structure,
$\Delta V/V$ = the volume change of the system, and
$\Delta\sigma'$ = the change in effective stress.

3.2 Principle of Effective Stress in Partially Saturated Nonswelling Soils

Due to the presence of a two-phase (compressible) pore fluid, and the interacting forces between the various phases, Terzaghi's effective equation, equation (1), becomes invalid. Earlier geomechanics researchers have attempted to modify this equation to account for the differential pressures in the pore fluid system. Of the various expressions defining effective stress, the one proposed by Bishop/3/ was considered the most general, since it included a term for the pressure in the gas phase. It was of the form:

$$\sigma' = \sigma - u_a + \chi * (u_a - u_w) \quad (3)$$

where u_a = pressure in the gaseous phase,
u_w = pressure in the liquid phase, and
χ = a parameter considered to be dependent mainly on the degree of saturation, S, but also on the soil structure and cycle of wetting, drying, or stress loading to a particular value of S/4/.

The above equation satisfies the boundary cases of a single pore fluid, ie, when the soil is saturated χ = 1; when the soil is dry χ = 0. Equation (3) may be rewritten in the form:

$$\sigma' = \sigma - \{ \chi * u_w + (1 - \chi) * u_a \} \quad (4)$$
$$= \sigma - \dot{u}$$

The quantity $\{ \chi * u_w + (1 - \chi) * u_a \}$ may be considered as an equivalent pore pressure. It was this form of the equation that attracted many research workers to restate the principle of effective stress in a form similar to the one proposed by Terzaghi for saturated soils.

Bishop et al/5/ showed that χ values obtained from drained triaxial tests on specimens of Braehead silt, placed as a slurry and drained by suction, compared favourably with theoretic values calculated by Donald/6/. It was shown that there was no simple general relationship for compacted soils and that the value of the parameter χ in the relationship between volume change and effective stress was not necessarily identical to that in the relationship between strength and effective stress.

Jennings and Burland/7/ stated that these tests, while showing that equation (3) was statically correct, could not be interpreted as demonstrating the validity of the principle of effective stress. To do this it was necessary to show that the soil behaviour was unaffected by changes in $(\sigma - u_a)$ and $(u_a - u_w)$ such that their sum was constant. On the basis of tests on a number of soils, Jennings and Burland/7/ demonstrated that equation (3) did not account for the behaviour of many soils below a critical degree of saturation. Consequently it was suggested that the stress defined by equation (3) should be called the intergranular stress σ'_i or the equivalent intergranular stress σ'_{ei}.

Bishop and Blight/8/ reexamined the principle of effective stress for both saturated and partially saturated soils. It was stated that in equation (3) a change in the value of the term $(u_a - u_w)$ did not correspond directly to a change in neutral stress, since the term represented a pressure difference attributable to surface tension acting, in general, over only a part of the surface area of the soil particles. A change in the value of $(u_a - u_w)$ was almost invariably accompanied by a significant change in the value of χ. Furthermore, the presence of large surface-tension forces within the soil led to differences in soil structure in samples following apparently similar effective-stress paths. These concepts implied that the value of χ was less unique than usually is assumed. For the case of compression under all-round pressure it was suggested by Bishop and Blight/8/ that the results could be expressed as a path in a space with the axes: void ratio, e, stress difference $(\sigma - u_a)$, and suction $(u_a - u_w)$. If the shape of the surface could be defined, a function of $(u_a - u_w)$ could be obtained and, for a specific stress path, the following equation could be evaluated/9/.

$$\sigma' = (\sigma - u_a) + f(u_a - u_w)$$

The validity of using the applied stress and suction to describe the deformation characteristic of a soil under all-round pressure and K_o loading were demonstrated by Matyas and Radhakrishna/9/. Subsequently,

Fredlund and Morganstern/10/ also proposed and experimentally tested constitutive relations for volume change in partially saturated soils, with the intergranular stress within a soil element expressed as/11/:

$$\sigma'_i = \sigma - u_a + a_w / a * (u_a - u_w) \qquad (5)$$

where a_w = cross-sectional area occupied by water,
a = total cross-sectional area,
u_a = pressure in air,
u_w = pressure in water,
σ = applied pressure, and
σ' = intergranular stress.

and the deformation of the soil determined by/10/:

$$\Delta V/V = m_1^s * \Delta(\sigma - u_a) + m_2^s * \Delta(u_a - u_w) \qquad (6)$$

where m_1^s = compressibility of soil structure when $\Delta(u_a - u_w)$ is zero, and
m_2^s = compressibility of soil structure when $\Delta(\sigma - u_a)$ is zero.

3.3 Principle of Effective Stress in Saturated Swelling Soils

For a highly swelling soil such as bentonite, the particles are very small, the specific surface is large, and ions in the pore fluid are absorbed on to the soil particle surfaces. The resulting electrical forces influence how the clay reacts to external stresses. Therefore equation (1) requires modifications to adequately describe the behaviour of this clay.

A bentonite contains a large amount of montmorillonite, which consists of two silica sheets bonded to a central alumina sheet. The interaction between the double layers around the clay particles creates a repulsive force R, and the attractive force is mainly caused by the van der Waals attractive forces, A between closely spaced neighbouring crystal structures. The attractive and repulsive electrical forces between the particles may be of the same order as contact stresses or body-weight forces/12/. The balance between the repulsive and attractive forces influences the separation between the clay particles, and hence the equilibrium void ratio of the clay under given stresses. Sridharan et al/13/ and Mitchell/11/

extended the application of equation (1) for soils containing expanding clay minerals at a water content (and void ratio) equilibrium:

$$\{\sigma\} = \{\sigma_i'\} + \{|R - A|\} + \{u(l)\} \quad (7)$$

Here $\{\sigma_i'\}$ is the intergranular pressure; $\{|R-A|\}$ is the swelling pressure attributable to electrochemical charge fields round the particles; and $\{u(l)\}$ is the pore-water pressure in the unbound water in the larger pore spaces.

3.4 Principle of Effective Stress for Partially Saturated Swelling Soils

Based on equations (5) and (6), the intergranular stress within a partially saturated swelling clay can be written as:

$$\sigma_i' = (\sigma - u_a) + a_w/a * (u_a - u_w) - (R - A) \quad (8)$$

Fredlund et al/14/ proposed a method to predict total heave of expansive clay. The stress state variables selected were $(\sigma - u_a)$ and $(u_a - u_w)$.

4.0 DEFORMATION OF SOILS UNDER A HYDRAULIC LOAD

With the decommissioning of the repository vault, the groundwater pressure around the shaft backfill will be redistributed, resulting in a large hydraulic pressure acting upon the shaft backfill at great depths. Compression of the partially saturated clay-based backfill will occur because of the large hydraulic load. Thus one of the functional requirements for the shaft backfill is that, with the ingress of water, the backfill should be able to expand and reseal the annular gap between the backfill and the shaft.

When a shaft backfill of swelling clay under hydraulic load comes in contact with water, the clay swells and a film of gel is formed around the perimeter of the shaft backfill. If this film of gel is compressed by the hydraulic load rendering it almost impervious, further ingress of water into the interior of the shaft backfill will be retarded, causing a decrease in the rate of swelling of the backfill. In this study, the occurrence of this phenomenon is termed "impervious skin effect."

The deformation behaviour of a clay-based shaft backfill with and without the impervious skin effect can be significantly different. This is so because the impervious skin effect alters the head prescribed hydraulic boundary condition induced by the applied hydraulic load at the shaft backfill and hydraulic load interface to an impervious hydraulic boundary condition.

In the following sections, the deformation mechanism of a shaft backfill subjected to a hydraulic load is examined. In addition, the implication of the occurrence of the impervious skin effect on the performance of a clay-based shaft backfill is also discussed.

4.1 Saturated Nonswelling Soils

The application of a vertical hydraulic load, P, on a radially confined saturated nonswelling soil element will induce an excessive pore-water pressure, u, in the soil element. If water is assumed to be incompressible within the range of applied stress, the induced pore-water pressure within the soil element is equal to the applied load. No hydraulic gradient exists between the pore-water pressure within the soil element and the applied hydraulic load. Thus the effective stress, equation (1), within the soil element remains unchanged and no deformation occurs in the backfill.

With the impervious skin effect, an impervious hydraulic boundary condition is imposed at the shaft backfill and hydraulic load interface. The increase in the pore-water pressure in the shaft backfill is equal to the applied hydraulic load and no deformation of the shaft backfill occurs.

4.2 Saturated Swelling Soils

As with the case above, with the application of a hydraulic load, the pore-water pressure induced in the soil element equals the applied hydraulic load. The intergranular stress, equation (7), within the soil element remains unchanged. If the swelling pressure within the soil element is not in equilibrium with the applied stress, the swelling clay will absorb water from the imposed hydraulic boundary. The soil will swell to an equilibrium position along the stress-state path A-A' shown in Figure 1. This equilibrium position lies on the swelling equilibrium line, SEL, defined by Graham et al/15/.

With the impervious skin effect, no water can enter the soil element. The stress state of the soil remains unchanged and no deformation of the soil element occurs.

4.3 Partially Saturated, Nonswelling Soils

The stress-state paths for a radially confined partially saturated nonswelling soil element subjected to a vertical hydraulic load with and without the impervious skin effect are plotted in Figure 2. Initially, the stress state of the soil element is at position A where the soil is subjected to an initial suction pressure. With the application of a hydraulic load, P, the soil deformed to position B due to the compressibility of the pore-air. For a partially saturated soil, the pore-air and pore-water pressures induced by the applied hydraulic load are less than the pressure of the applied hydraulic load. Water ingresses into the soil due to the induced hydraulic gradient. With time, the ingressed water fills the larger voids of the soil element and destroys the menisci (or suction) within the soil. During this process, the stress-state within the soil element moves from points B to D (ie, rebound). Simultaneously, the pore-air, which is under high pressure induced by the hydraulic load, dissolves into the water. With further ingress of water the pore-water pressure within the soil element increases and equilibrates at a value equal to the applied hydraulic load (from stress-state point D to point E shown in Figure 2). The $(\sigma - u_a)$ term in equation (5) becomes zero. This change in stress-state indicates the rebound (or expansion) of the soil element due to the reduction in the intergranular stress within the soil element.

With the impervious skin effect on the shaft backfill, water from the hydraulic load cannot flow into the soil. The soil compresses from stress-state points A to B. No further compression of the soil can occur.

4.4 Partially Saturated Swelling Soils

The deformation behaviour of a radially confined partially saturated swelling soil element subjected to a vertical hydraulic load, P, is shown in Figure 3. With the application of a vertical hydraulic load to the soil element, immediate compression of the soil element occurs due to the compression of the pore-air space within the soil and the load induces a pore-air pressure of u_a (from stress-state point A to point B).

Since the hydraulic load has a larger value than the induced pore-air pressure within the soil element, ingress of water into the soil element occurs. With time, water fills the larger voids within the soil element and destroys the menisci in the soil element. If none of the water in the larger voids is being absorbed by the double layer of the clay particles, the change in the stress-state of the soil element follows path BC. If part of the water in the larger voids is absorbed by the double layer of the clay particles, resulting in the reduction of swelling pressure of the soil, (R - A) in equation (8), then the change in the stress state of the soil follows the path BD. If the water in the larger voids is being absorbed instantaneously, then the change

in the stress state of the soil element follows path BE. During the ingress of water into the larger voids of the soil element, the pore-air, which is great pressure, dissolves into the water.

With further ingress of water into the soil element, the soil will expand because of the reduction of the swelling pressure, (R - A), and the stress, $(\sigma - u_a)$, shown in equation (8). The soil expands continuously until the stress-state of the soil element reaches point F as shown in Figure 3. When the impervious skin effect occurs, the backfill compresses and the change in the stress-state of the soil terminates at point B.

The above discussion indicates that the occurrence of the impervious skin affect would destroy the resealing function of shaft backfill. Considering the occurrence of the impervious skin effect, the concept of using geological materials as shaft seal should be reevaluated.

5.0 THE PROPOSED ANALYTICAL MODEL

The stress-state change of a partially saturated swelling soil subjected to a hydraulic load is shown in Figure 3. To predict the time-dependent deformation of a clay-based backfill caused by the application of a hydraulic load, the stress-state path (BC/BD/BE) of the backfill during the release of the suction pressure within the backfill has to be defined. In this study it is assumed that the rate at which the double layers of clay particles absorb water from the larger voids of the backfill is faster than the rate of water ingress into those voids. Thus the stress-state path of the soil element is defined by line BE shown in Figure 3. This assumption supports the observation that bentonite powder swells instantaneously when free water is available. The second assumption is that under the range of applied hydraulic load the pore-air dissolves completely into the pore-water.

The time-dependent deformation of a clay-based shaft backfill under a constant hydraulic load is governed by the change in pore-air pressure and pore-water pressure within the backfill. With the determination of the time-dependent pore-air pressure and pore-water pressure distributions within the backfill, the time-dependent deformation of the backfill can be calculated. The development of the analytical model for the prediction of the time-dependent pore-air and pore-water pressures within a shaft backfill subjected to large hydraulic load is presented here.

5.1 Analytical Model

Continuity of a partially saturated soil element requires that the overall volume change of the element must equal to the sum of the volume changes associated with the component phases/16/. If the soil particles are incompressible, the continuity requirement can be written as:

$$\Delta V/V = \Delta V_w/V + \Delta V_a/V$$

where V = overall volume of the soil element,
V_w = volume of water in the soil element, and
V_a = volume of air in the soil element.

Along the stress-state path BE, deformation of the backfill occurs due to the reduction in suction, $(u_a - u_w)$, and swelling pressure (R - A) within the backfill. The one-dimensional equation governing the change in suction with time can be developed based on the mass balance of flow.

The volume of water entering and leaving the soil element shown in Figure 4 is described by Darcy's law.
Flow into the bottom of the element:

$$q_y = k_y * (-\partial h/\partial y) * dx * dz \qquad (9)$$

Flow out of the top of the element

$$q_y = \{(k_y + \partial k_y/\partial y * dy) * (-\partial h/\partial y - \partial^2 h/\partial y^2 * dy)\} * dx * dz \qquad (10)$$

where k_y = hydraulic conductivity in the Y direction, and
h = total head.

The net water flux into the element from vertical flow is expressed as:

$$\{(k_y + \partial k_y/\partial y * dy) * (-\partial h/\partial y - \partial^2 h/\partial y^2 * dy)\} * dx * dz - k_y * (-\partial h/\partial y) * dx * dz \qquad (11)$$

For the condition of constant hydraulic conductivity, $\partial k_y/\partial y = 0$, equation (11) can be simplified to:

$$-(k_y * \partial^2 h/\partial y^2) * dy * dx * dz \qquad (12)$$

The total head, h, can be expressed

$$h = u_w/r_w + Y \qquad (13)$$

where u_w = pore-water pressure,

γ_w = density of water, and

Y = elevation head.

Substitute equation (13) into equation (12). The net flux of water is

$-(k_y/\gamma_w * \partial^2 u_w/\partial y^2) * dy * dx * dz$, therefore,

$\partial(\Delta V_w)/\partial t = -(k_y/\gamma_w * \partial^2 u_w/\partial y^2) * dy * dx * dz$, or

$$\partial(\Delta V_w/V)/\partial t = -(k_y/\gamma_w * \partial^2 u_w/\partial y^2) \quad (14)$$

Assuming the rate of diffusion is negligible, the hydraulic boundary condition imposes an impermeable boundary for the pore-air within the soil element. The change in pore-air volume within the soil element is controlled by (1) the compressibility, and (2) the solubility of the pore-air. In view of the complexity of the process, and also as the initial trial to predict the time-dependent deformation behaviour of the backfill, it is assumed that within the practical range of pressure, the pore-air will dissolve completely in the pore water. With time, the air void will be filled with water influx from the hydraulic boundary. Thus the change in volume of the soil element shown in Figure 4 can be written as:

$$\Delta V/V + \Delta V_a/V = \Delta V_w/V \quad (15)$$

The constitutive relations proposed by Fredlund and Morgenstern/10/ to link the stress- and deformation-state variables for both soil structure and water phase are:

$$\begin{aligned}\Delta V/V &= m_1^{\circ} * d(\sigma - u_a) + m_2^{\circ} * d(u_a - u_w) \\ &= m_1^{\circ} * d(\sigma - u_a) + m_2^{\circ} * d(u_a) + m_2^{\circ} * d(-u_w)\end{aligned} \quad (16)$$

where σ = external applied hydraulic load,

u_a = pore-air pressure,

u_w = pore-water pressure,

m_1° = compressibility of the soil structure when $d(u_a - u_w)$ is zero, and

m_2° = compressibility of the soil structure when $d(\sigma - u_a)$ is zero. In this study, the volume change of the soil structure due to the reduction in swelling pressure is being implicitly considered by this parameter.

Along the stress-state path BE, the applied hydraulic load, σ, and $(\sigma - u_a)$ are constant and equation (16) can be simplified to:

$$\Delta V/V = -m_1^{\circ} * d(u_w)$$

The rate of volume change is equal to:

$$\partial(\Delta V/V)/\partial t = -m_1^{\circ} * \partial(u_w)/\partial t \qquad (17)$$

Similarly, $\Delta V_w/V$ can be rewritten as:

$$- m_1^{\circ} * d(u_w)$$

The rate of air-volume change is equal to:

$$\partial(\Delta V_w/V)/\partial t = - m_1^{\circ} * \partial(u_w)/\partial t \qquad (18)$$

where m_1° = ratio of the volume of pore-air/volume of the soil element to the total suction within the soil element.

Substitute equations (17), (18), and (14) into equation (15)

$$-m_1^{\circ} * \partial(u_w)/\partial t - m_2^{\circ} \partial(u_w)/\partial t = - k_y/\gamma_w * \partial^2 u_w/\partial y^2 \qquad (19)$$

$$(m_1^{\circ} + m_2^{\circ}) \partial(u_w)/\partial t = k_y/\gamma_w * \partial^2 u_w/\partial y^2$$

Assume $c_v^{\circ} = k_y/\gamma_w /(m_1^{\circ} + m_2^{\circ})$ then equation (19) can be rewritten as:

$$\partial(u_w)/\partial t = c_v^{\circ} * \partial^2 u_w/\partial y^2 \qquad (20)$$

With the determination of the soil properties, m_2°, k_y and m_2°, the time-dependent deformation of the clay-based backfill caused by the reduction in suction and swelling pressure can be calculated. The k_y is measured with hydraulic conductivity tests, m_2° is equal to the slope of line BE (Figure 3) and m_2° is

assumed to equal the ratio of the volume of pore-air/volume of the soil element to the total suction within the soil element.

Along the stress-state path EF, the soil is saturated. Thus equation (20) can be written as:

$$m_2^s \, \partial (u_w)/\partial t = k_y/\gamma_w * \partial^2 u_w/\partial y^2 \qquad (21)$$

where m_2^s = slope of the swelling equilibrium line (SEL).

Equation (21) can be rewritten to:

$$\partial (u_w)/\partial t = c_v * \partial^2 u_w/\partial y^2 \qquad (22)$$

where $c_v = k_y/\gamma_w /m_2^s$

In an oedometer test, the compression of the partially saturated clay-based backfill at different applied load can be measured (stress-state path AB in Figure 3). Stress-state path EF is determined by performing an oedometer test on the backfill specimen with initial moisture content at its liquid limit. The coefficient of consolidation, c_v, of the shaft backfill at different stress levels is determined with the standard oedometer test procedures. The parameter c_v^u along the stress-state path BE is calculated with the partially saturated hydraulic conductivity, k_y, of the backfill, and the parameter (m_2^s + m_2^a).

A one-dimensional computer code based on the integrated finite difference technique is developed to solve the equations (20) and (22). The computer code is listed in Appendix A. The change in material properties of the backfill caused by the change in stress state within the backfill is accounted for in the simulation. With the determination of the change in pore-water pressures with time, the deformation of the soil domain along stress-state paths AB, BE, and EF can be predicted.

5.2 Calibration of the Analytical Model with Preliminary Laboratory Data

5.2.1 The One-Dimensional Compression Test

The measured deformation-versus-time data from a one-dimensional compression test conducted in the high-pressure compression cell is used to assess the validity of the analytical model. The clay-based shaft

backfill mixture used in this study consisted of 25% by weight of Lake Agazzi clay and 75% by weight of a granular material provided by Atomic Energy of Canada Limited/17/. The grain-size distribution of the backfill mixture is plotted in Figure 5. Because of the restriction in the size of the test equipment, the backfill mixture having a particle size greater than sieve #40 was discarded.

The properties of the backfill specimen required by the analytical model were determined by oedometer tests. The results are summarized and presented in Appendix B. The resulting deformation characteristics of the backfill are plotted in Figure 6. The measured deformation-versus-time data of a sample under an instantaneous hydraulic load of 3.5 MPa in the high-pressure compression cell are plotted in Figure 7.

5.2.2 The Analytical Prediction

The one-dimensional model used to simulate the one-dimensional compression test described above is presented in Figure 8. The 30-mm thick soil specimen used in the compression test is discretized, in the vertical direction, into 30 elements of equal thickness. The hydraulic load applied in the compression test is simulated with a head prescribed boundary condition at the top of the model. At the bottom of the model, a no-flow boundary condition is used to simulate the impervious boundary at the bottom of the compression cell. The pressure-dependent compressibility and hydraulic conductivity properties of the backfill as included in Appendix B are used in the analysis.

During the analysis, the calculated pore-water pressure at the nodal point of each element is output at regular time intervals. Based on the calculated pore-water pressure distribution and the pressure-dependent deformation properties of the backfill (Appendix B), the deformation of each element is calculated. The total deformation of the specimen at any time is equal to the sum of the deformations that occur in each element. The predicted deformation-versus-time results are plotted in Figure 9.

5.2.3 Comparison of Results

Based on the material properties determined from the oedometer tests, and the applied load condition, the predicted deformation-versus-time results are plotted in Figure 10. The data indicate that initially, the analytical model underpredicted the deformation of the soil. When the deformation of the soil approaches the steady state, the model overpredicted the deformation of the soil by about 10%. The final moisture contents at the upper, middle, and lower levels of the specimen are 31.2%, 27.5%, and 23.2%, respectively. The corresponding moisture content values predicted by the analytical model are 32.3%, 29.1%, and 26.4%.

The good comparison between the predicted and measured deformation of the backfill specimen, and between the predicted and measured final moisture content of the backfill specimen provide tentative evidence of the general adequacy of the analytical model. The model will be further calibrated with additional data from the high-pressure compression cell.

6.0 CONCLUSIONS

Because of the inherent scale effect, the deformation behaviour of shaft backfill subjected to a hydraulic load cannot be adequately assessed by laboratory scale experiment. In this report, an analytical model is developed to assist in the assessment of the performance of a clay-based shaft backfill. This analytical model can also be used to assist the selection of a clay-based backfill mixture.

The time-dependent deformation of the shaft backfill depends significantly on the load applied to the backfill. It has been demonstrated that if a film of the backfill mixture at the shaft backfill-hydraulic load interface becomes impervious, compression of the shaft backfill will allow a short circuit path to develop. This concept is termed as the impervious skin effect.

The experimental results from the high-pressure compression tests performed on the clay-based shaft backfill and the results predicted form the analytical model indicate that with the ingress of water into the soil, a layer of the soil will become saturated. However, because of the large hydraulic gradient existing across the layer, water flows into the layer, resulting in a build-up of high pore-water pressure in the layer. This pore water pressure build-up will reduce the intergranular stress in the backfill. Such reduction in intergranular stress permits the layer to expand, thereby causing a positive contact stress between the shaft and the backfill. Such an expansion of the sample invalidates the concept of the impervious skin effect.

The comparison made, based on the limited predicted and measured results, indicates that the simplified analytical model developed may be used to predict the one-dimensional time-dependent deformation of shaft backfill. The good comparison in results also substantiates the assumptions made in the development of the model. Because of the limited data available in this study, further validation of the analytical model is in progress.

ACKNOWLEDGEMENTS

Mr. Bob Watts and Mr. Greg Loewen of the Civil Research Department performed most of the compression tests. Dr. H.S. Radhakrishna provided valuable comments and suggestions. Their contributions are gratefully acknowledged. The project was partially funded by Ontario Hydro's Nuclear Engineering Department.

Approved by:

for J.B. Brown
Manager
Civil Research Department

Prepared by:

K.C. Lau
Engineer
Soil Sciences Section

H.T. Chan
Senior Engineer
Soil Sciences Section

Reviewed by:

T.I. Carmichael
Supervising Engineer
Soil Sciences Section

KCL/HTC:LM

REFERENCES

(1) Chan, H.T. and K.C. Lau, "Shaft Backfill Study: High-Pressure Compression Cell and Preliminary Data", Ontario Hydro Research Division Report No. 90-129-K, September 13, 1990.

(2) Terzaghi, K., "The Shearing Resistance of Saturated Soils and the Angle Between the Planes of Shear", Proceedings of the First International Conference on Soil Mech and Found, England, Vol 1, 1936.

(3) Bishop, A., "The Principle of Effective Stress", Teknisk Ukebland, Vol 39, 1959.

(4) Bishop, A.W., "The measurement of Pore Pressure in the Triaxial Test", Proceedings of the Conference on Pore Pressure and Suction in Soils, London, 1960.

(5) Bishop, A.W., I. Alpan, G.E. Blight and I.B. Donald, "Factors Controlling the Strength of Partially Saturated Cohesive Soils", Proceedings of American Society of Civil Engineers, Research Conference on Shear Strength of Cohesive Soils, 1960.

(6) Donald, I.B., "The Mechanical Properties of Saturated and Partly Saturated Soils with Particular Reference to the Effects of Negative Pore-Water Pressures", Ph. D. Thesis, University of London, 1961.

(7) Jennings, J.E. and J.B. Burland, "Limitations to the Use of Effective Stress in Partly Saturated Soils", Geotechnique, Vol 12, 1962.

(8) Bishop, A.W. and G.E. Blight, "Some Aspects of Effective Stress in Saturated and Partly Saturated Soils", Geotechnique, Vol 13, 1963.

(9) Matyas, E.L. and H.S. Radhakrishna, "Volume Change Characteristics of Partially Saturated Soils", Geotechnique, Vol 18, 1968.

(10) Fredlund, D.G. and N.R. Morgenstern, "Constitutive Relations for Volume Change in Unsaturated Soils", Canadian Geotechnical Journal, Vol 13, 1976.

(11) Mitchell, J.K., "Fundamentals of Soil Behaviour", John Wiley and Sons, New York, 1976.

(12) Morgenstern, N.R. and B.I. Balasubramonian, "Effects of Pore Fluid on the Swelling of Clay-Shale", Proceedings of Fourth International Conference on Expansive Soils, Denver, Colorado, 1980.

(13) Sridharan, A. and G.V. Rao, "Mechanisms Controlling Volume Change of Saturated Clays and the Role of the Effective Stress Concept", Geotechnique, Vol 23, 1973.

(14) Fredlund, D.G., J.U. Hasan and H.L. Filson, "The Prediction of Total Heave", Proceedings of the Fourth International Conference on Expansive Soils, Denver, Colorado, USA, 1980.

(15) Graham, J., F. Saadat, M.N. Gray, D.A. Dixon and Q.Y. Zhang, "Strength and Volume Change Behaviour of a Sand-Bentonite Mixture", Canadian Geotechnical Journal, Vol 26, 1989.

(16) Fredlund, D.G. and J.U. Hasan, "One-Dimensional Consolidation Theory: Unsaturated Soils", Canadian Geotechnical Journal, Vol 16, 1979.

(17) Yong, R.N., P. Boonsinsuk, G. Wong, X.D. Ming, F. Caporuscio and P. Lytle, "Backfill Formulations for a Nuclear Fuel Waste Disposal Vault", Atomic Energy of Canada Limited Report No. AECL-9071, 1987.

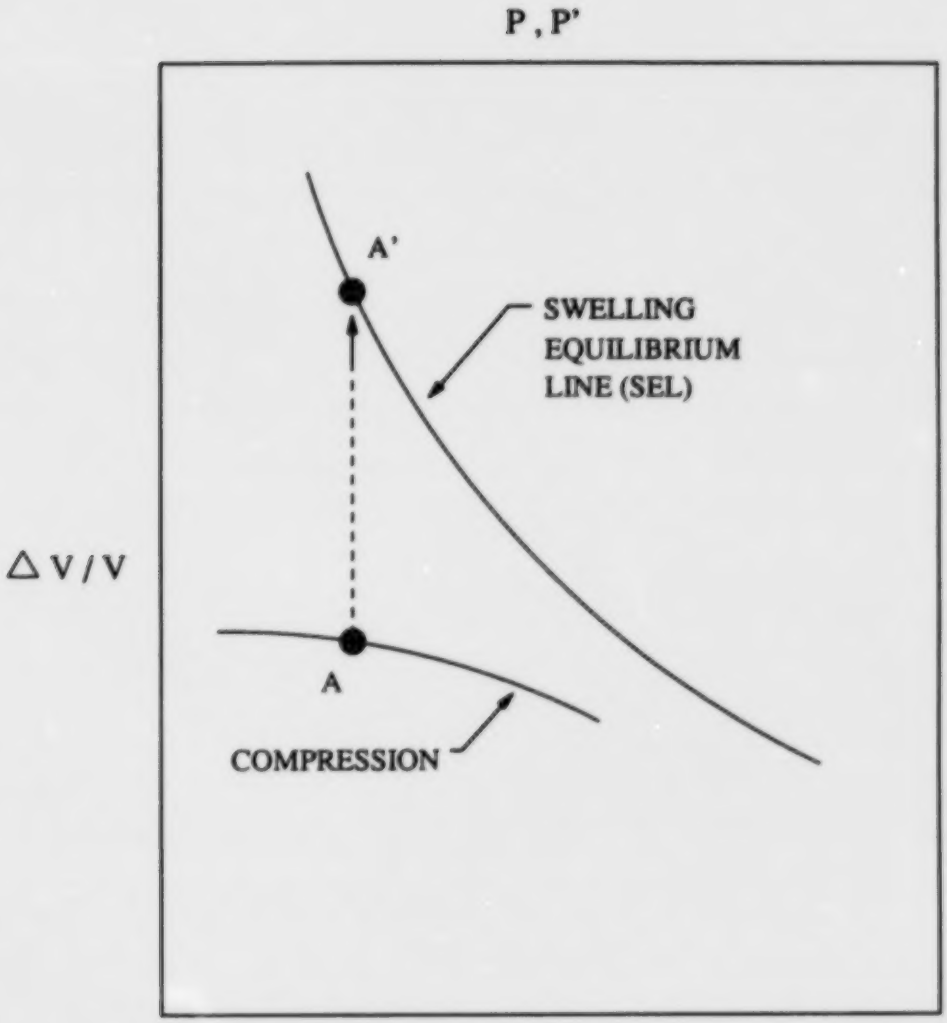

FIGURE 1 STRESS-STATE PATH OF SATURATED SWELLING SOIL.

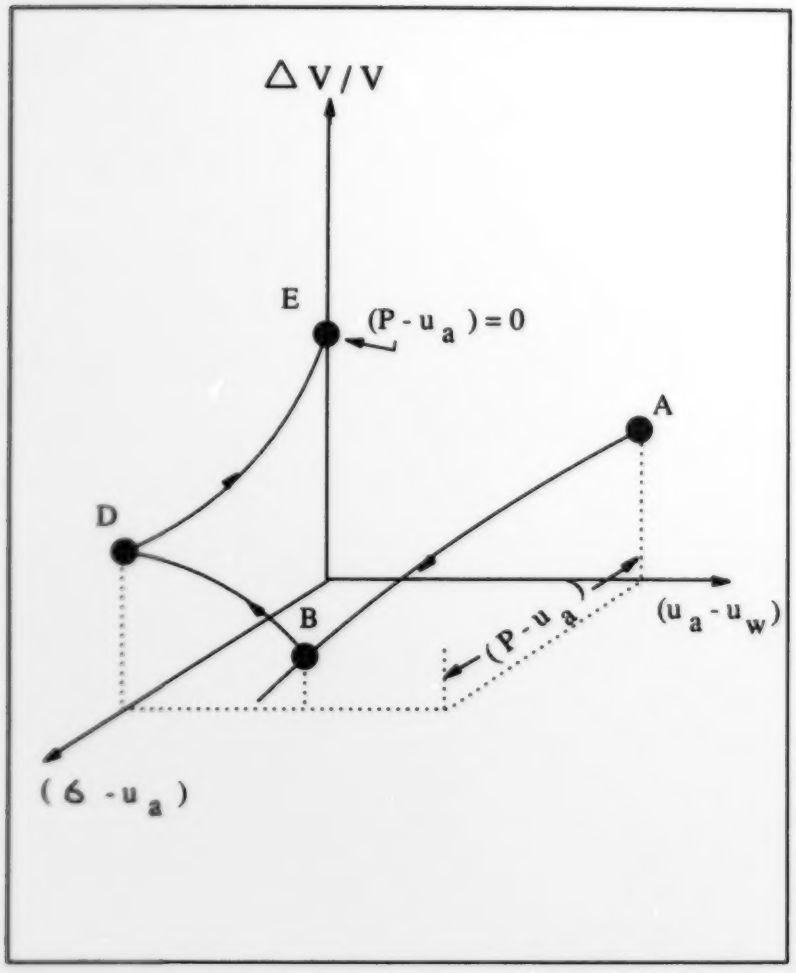

FIGURE 2 STRESS-STATE PATH OF PARTIALLY SATURATED NONSWELLING SOIL.

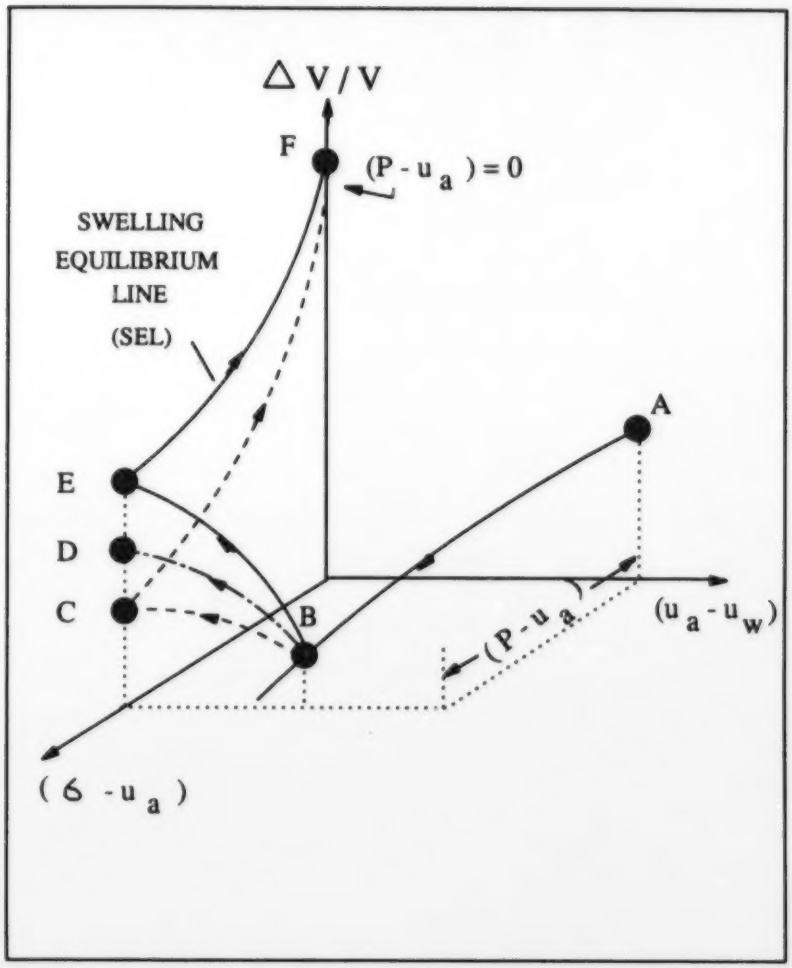

FIGURE 3 STRESS-STATE PATH OF PARTIALLY SATURATED SWELLING SOIL.

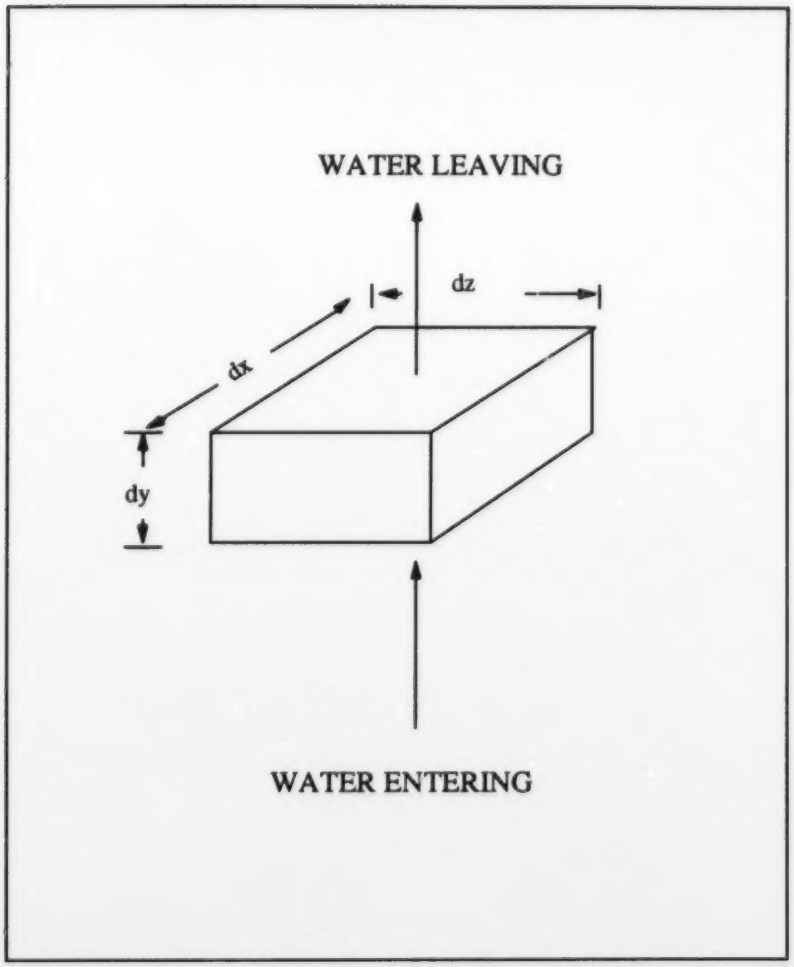

FIGURE 4 THE WATER BALANCE WITHIN A SOIL ELEMENT.

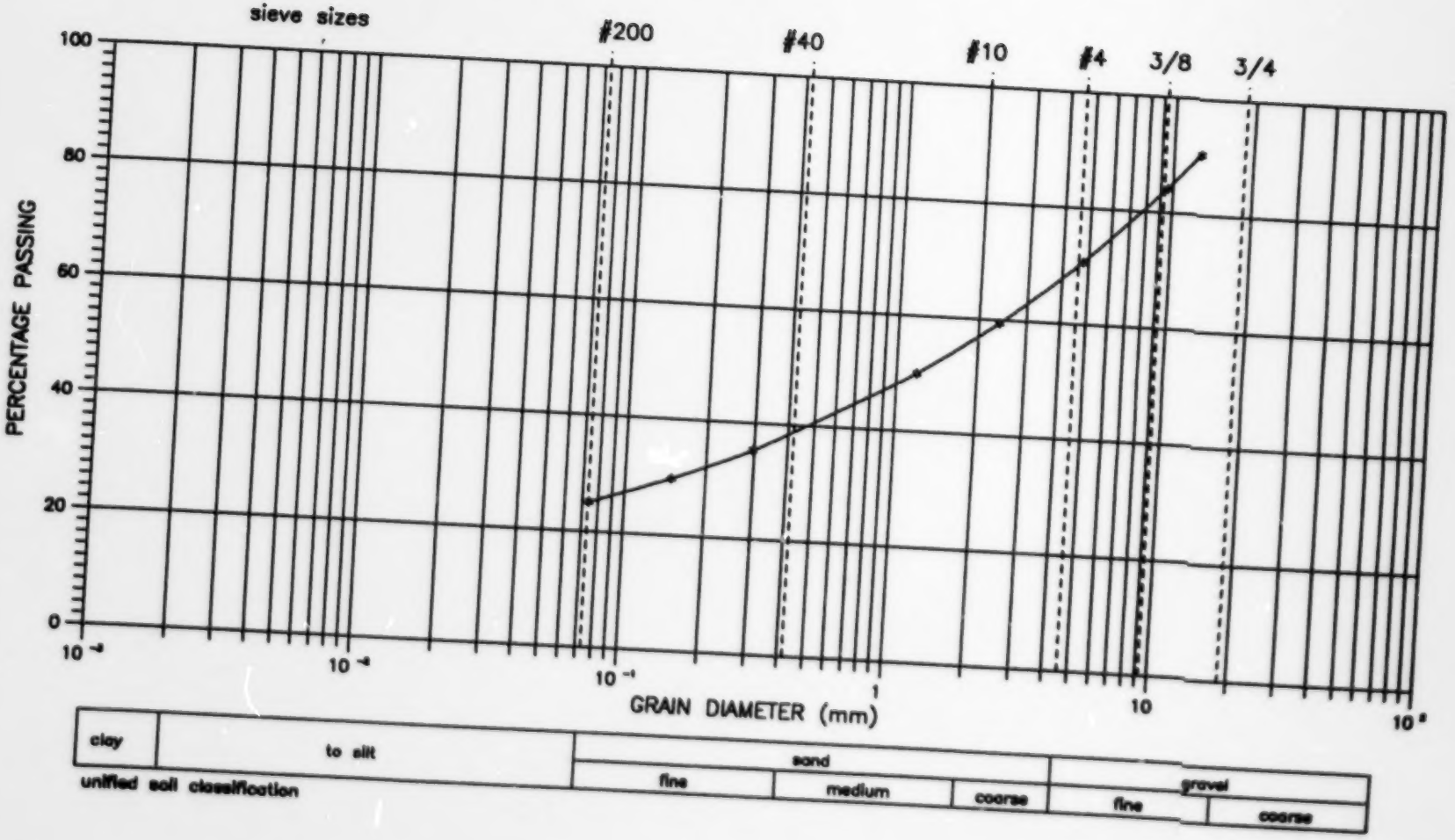

FIGURE 5 GRAIN SIZE DISTRIBUTION OF THE BACKFILL MIXTURE.

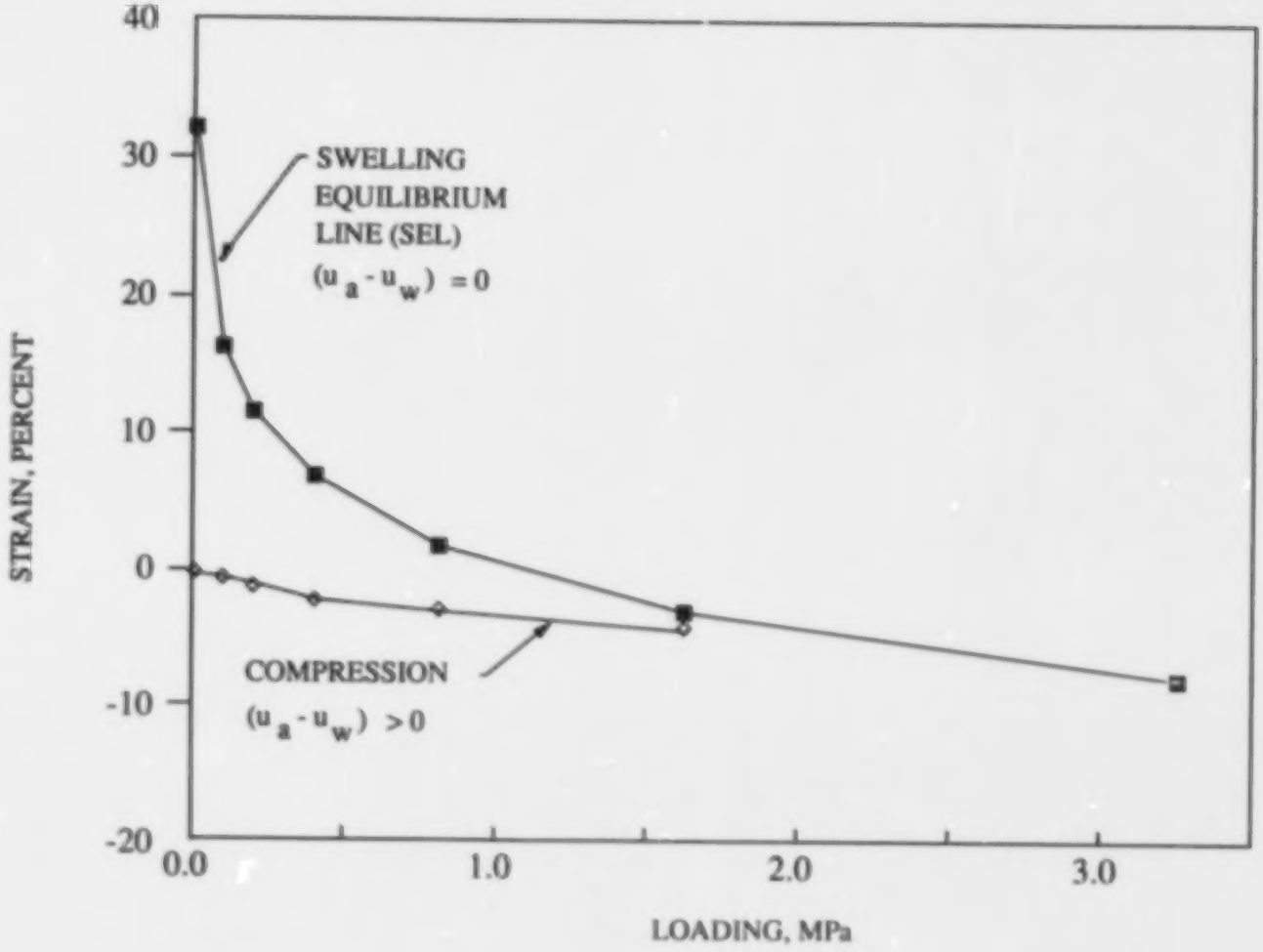

FIGURE 6 THE DEFORMATION PROPERTIES OF THE BACKFILL MATERIAL.

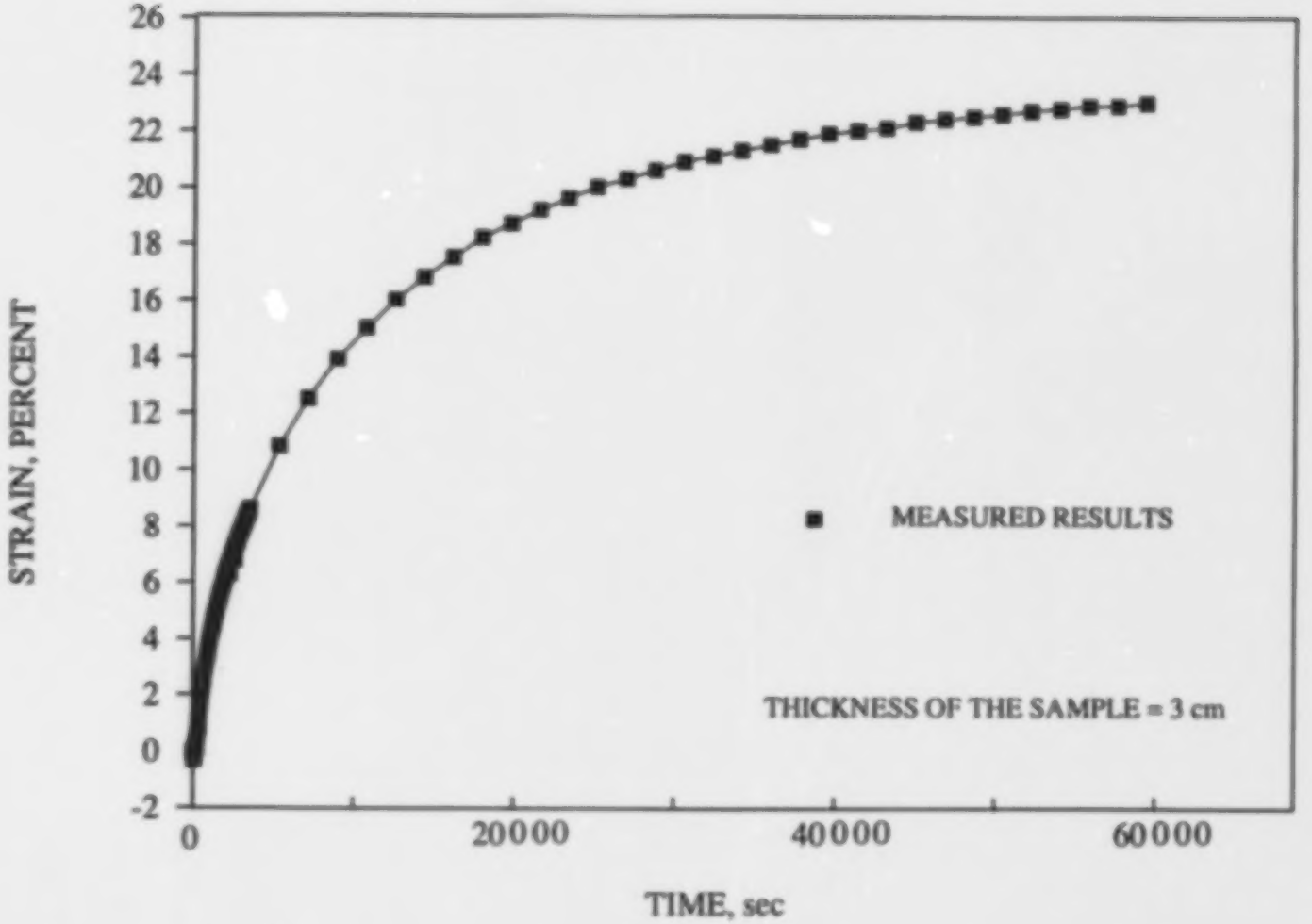

FIGURE 7 THE TIME-DEPENDENT DEFORMATION OF THE BACKFILL -- RESULTS FROM HIGH-PRESSURE COMPRESSION TEST.

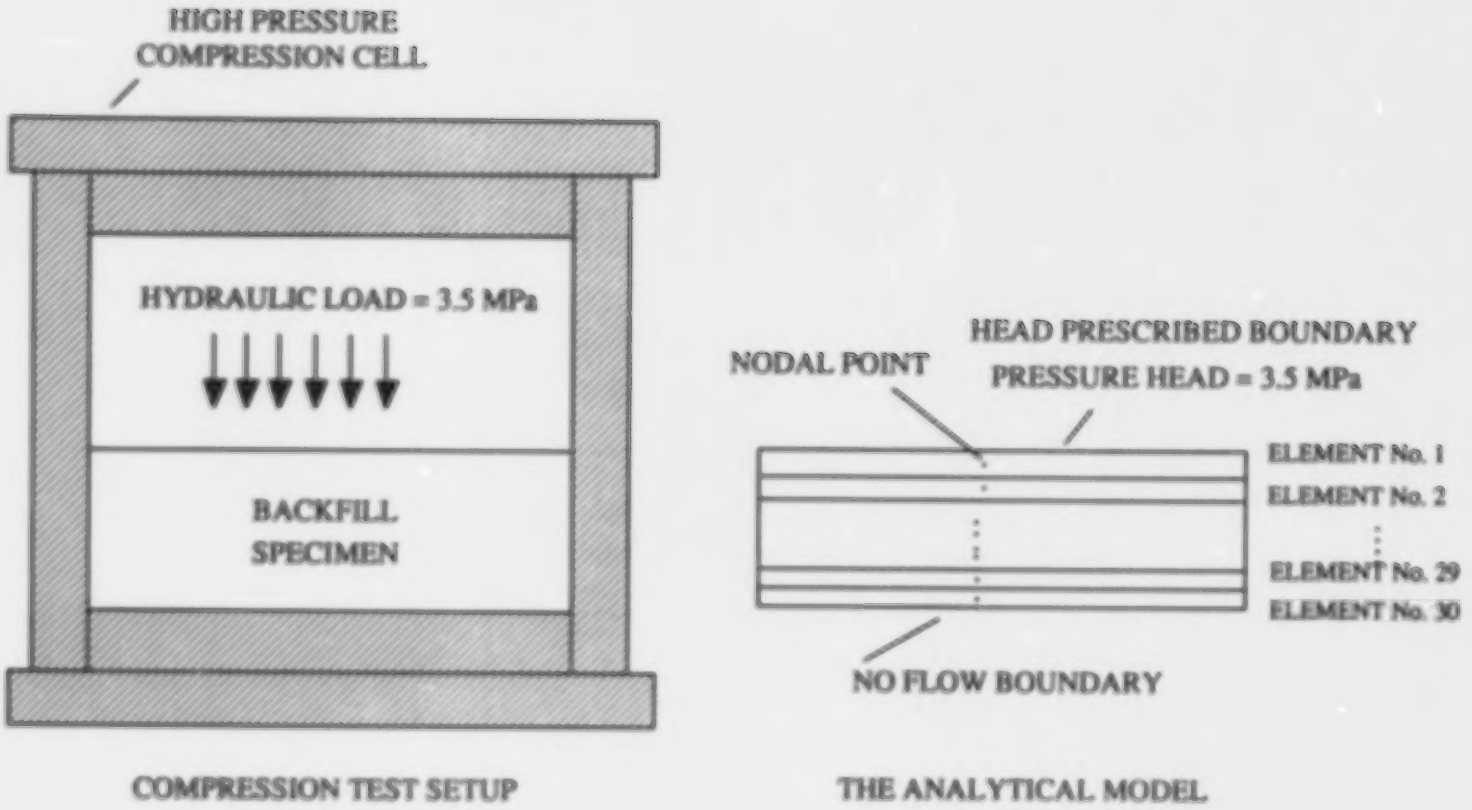

FIGURE 8 THE ANALYTICAL MODEL USED TO SIMULATE THE COMPRESSION TEST.

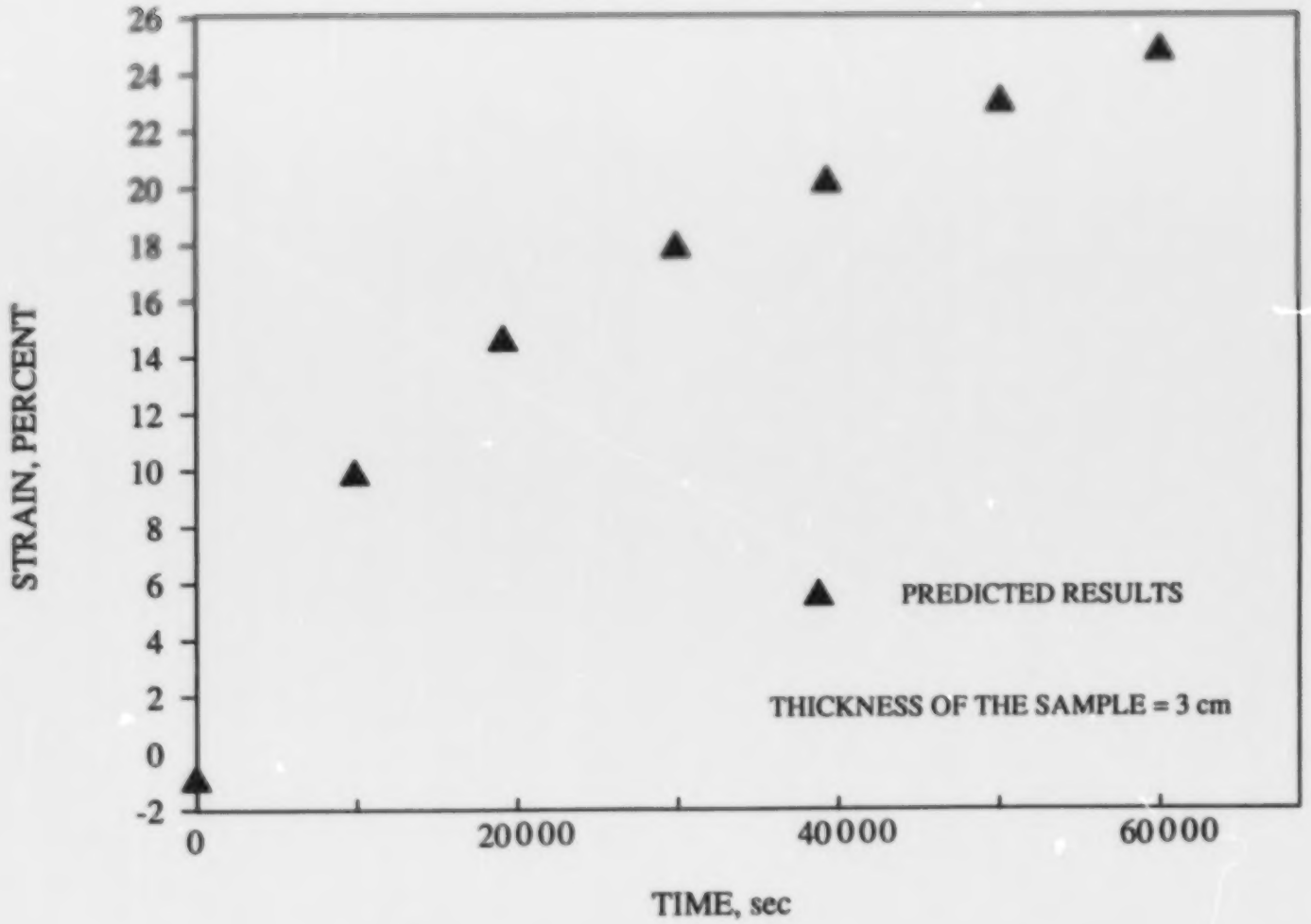

FIGURE 9 THE TIME-DEPENDENT DEFORMATION OF THE BACKFILL -- RESULTS FROM ANALYTICAL SIMULATION.

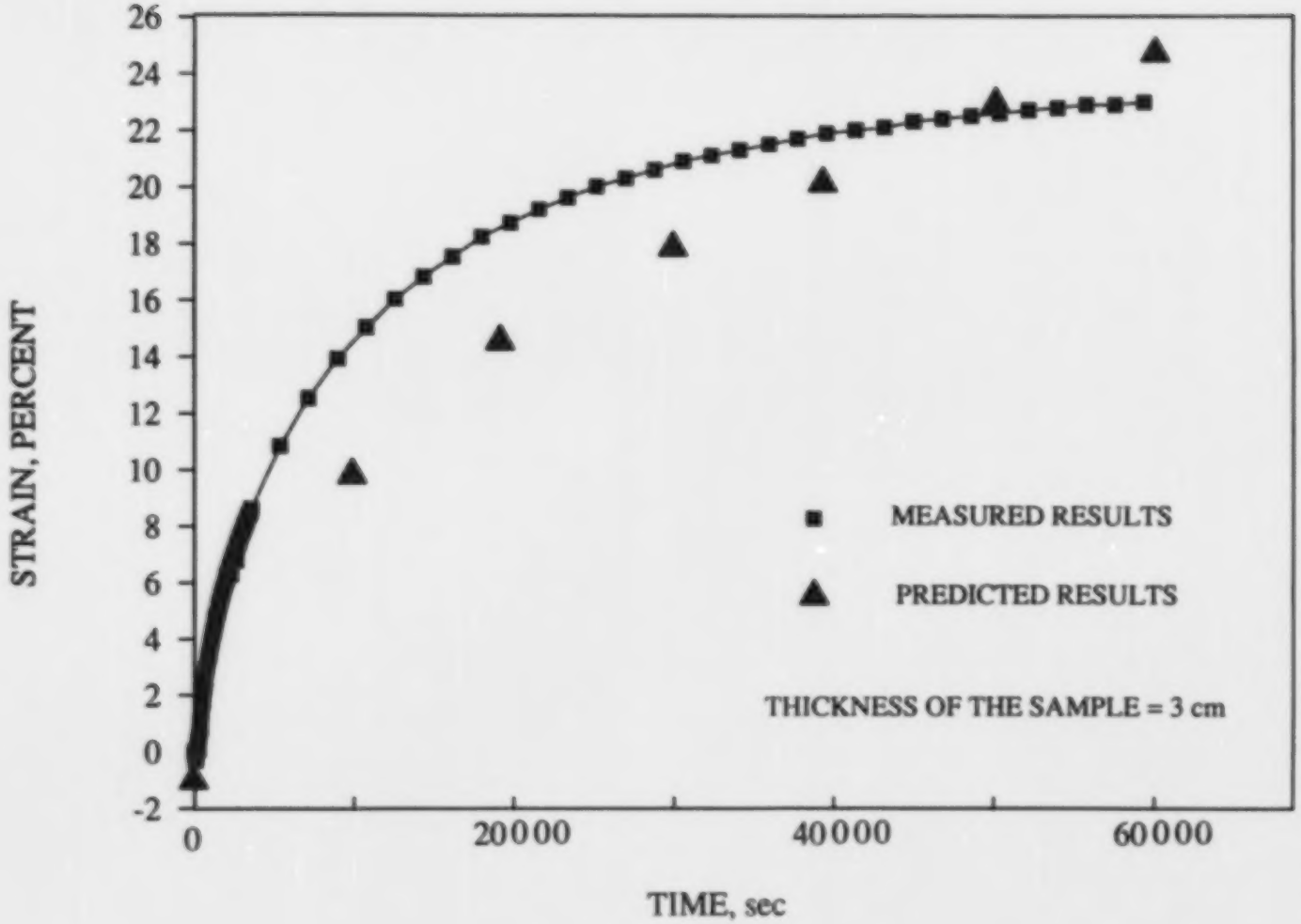

FIGURE 10 THE PREDICTED AND MEASURED TIME-DEPENDENT DEFORMATION OF THE BACKFILL SPECIMEN.

APPENDIX A

COMPUTER CODE LISTING

```
C
      COMMON MATL(6),MATAB(6,10),AMX(10,8,6),
     1 AMY(10,8,6),SLOMAT(10,8,6),KCON(6,3)
      COMMON ATCAL(5500,8),NODMAT(5500),NOXMAT(5500),TEMP(5500),
     1 PHI(5500),DTEMP(5500),DPHI(5500),DTF(5500),DMF(5500)
      COMMON VOL(5500),Z(5500),NTYPE(5500),NODE(5500)
      COMMON NOTE(5500),NOXE(5500)
      COMMON NOD1(15600),NOD2(15600),NOX1(15600),NOX2(15600),
     1 DEL1(15600),DEL2(15600),AREA(15600),KODCON(15600)
      COMMON NODS(200),NODSB(200),NOXS(200),NOXSB(200),
     1 HTSURT(20,200),HMSURT(20,200),SMLOH(20,200),STLOH(20,200),
     1 HTSURE(200),HMSURE(200),LTABH(200),AREAS(200),
     1 TMVARH(20,200),TTVARH(20,200)
      COMMON PHIB(20,200),TEMPB(20,200),NODB(200),ZB(200),
     1 LTABT(200),STLOT(20,200),SMLOT(20,200)
      COMMON GMT(20,200),GM(5500),NMOXG(200),NMODG(200),
     1 TMVARG(20,200),LMTABG(200),SMLOG(20,200)
      COMMON GTT(20,200),GT(5500),NTOXG(200),NTODG(200),
     1 TTVARG(20,200),LTTABG(200),STLOG(20,200),
     1 DDPHI(5500),DDTEMP(5500),ETRORX(5500),EMRORX(5500)
      COMMON ETROR(5500),EMROR(5500)
      COMMON TRAN(15600,5),ZIP(5500,4)
      COMMON FTS(200),FMS(200),DTFS(200),DMFS(200),FTB(200),
     1 FMB(200),DMFB(200),DTFB(200),TTRANS(200),TMRANS(200),
     1 TSIB(200),PSIB(200),TSIBS(200),PSIBS(200)
      COMMON SMLIM(5500),STLIM(5500),TIMEB(20,200)
      COMMON NMAT,NODES,KCYC,MAX,MID,SET,NOSPEC,NOCON,
     1 NVARH,NOSCON,NODBS,NVART,NMOGEN,NMVARG,
     1 NTOGEN,NTVARG
      COMMON SUMTIM,FOR,FORD,DELTS,DELT,DPIMAX,MAXITT,
     1 SPEED,NUTX,NUTSUM,NUTS,NUTMAX,TAU,S1,S2
      COMMON NUTAVG,PHIVRY,TEMVRY,SMALL,SMALT,BIG,KWTT,
     1 KDATA,KOOOD,MS,MSS,MSF,DELTMX,DELTO,NREG,DELTG
      COMMON RATTG,RATMG,RATT1,RATM1,RATT2, RATM2,TIMAX,DELTSS,
     1 MCYC,KSPEC,IPRINT,TIMEP
      COMMON NIT
      COMMON/SEEY/INPUT1,INPUT2,NOW
        CHARACTER*4 AANAME(19)
        INPUT1=5
        INPUT2=6
C
      READ(INPUT1,100) AANAME
100    FORMAT(1X,19A4)
      WRITE(INPUT2,100) AANAME
      CALL LINE
      KCYC=-1
      KWTT=0
      NSTOP=0
      NOW=0
C
C
      CALL CONTL
      CALL LINE
      CALL MATINP
      CALL LINE
      CALL ELEM
      CALL LINE
      CALL ELECON
      CALL LINE
```

```
      CALL SBCON
      CALL LINE
      CALL SURF
      CALL LINE
      CALL LINE
      CALL MGEN
      CALL LINE
      CALL INICON
      CALL LINE
      CALL INTST
      CALL LINE
C
C
C
1     KCYC=KCYC+1
2     CALL MATCAL
      IF((NMOGEN+NTOGEN).NE.0) CALL SOURCE
      IF(NOCON.NE.0) CALL ESOLVE
      IF(NOSCON.NE.0) CALL BSCAL
      IF(NOSPEC.NE.0) CALL ISOLVE
      IF(NSTOP.NE.0) GO TO 5
4     CALL TIME
      IF(KWIT) 2,1,3
3     IF((KWIT-5)*(KWIT-9).EQ.0) GO TO 5
      NSTOP=1
      GO TO 1
5     CONTINUE
      STOP
        END
      SUBROUTINE CONTL
C
C
C
C
      COMMON MATL(6),MATAB(6,10),AMX(10,8,6),
     1 AMY(10,8,6),SLOMAT(10,8,6) KCON(6,3)
      COMMON ATCAL(5500,8),NODMAT(5500),NOXMAT(5500),TEMP(5500),
     1 PHI(5500),DTEMP(5500),DPHI(5500),DTF(5500),DMF(5500)
      COMMON VOL(5500),Z(5500),NTYPE(5500),NODE(5500)
      COMMON NOTE(5500),NOXE(5500)
      COMMON NOD1(15600),NOD2(15600),NOX1(15600),NOX2(15600),
     1 DEL1(15600),DEL2(15600),AREA(15600),KODCON(15600)
      COMMON NODS(200),NODSB(200),NOXS(200),NOXSB(200),
     1 HTSURT(20,200),HMSURT(20,200),SMLOH(20,200),STLOH(20,200),
     1 HTSURE(200),HMSURE(200),LTABH(200),AREAS(200),
     1 TMVARH(20,200),TTVARH(20,200)
      COMMON PHIB(20,200),TEMPB(20,200),NODB(200),ZB(200),
     1 LTABT(200),STLOT(20,200),SMLOT(20,200)
      COMMON GMT(20,200),GM(5500),NMOXG(200),NMODG(200),
     1 TMVARG(20,200),LMTABG(200),SMLOG(20,200)
      COMMON GTT(20,200),GT(5500),NTOXG(200),NTODG(200),
     1 TTVARG(20,200),LTTABG(200),STLOG(20,200),
     1 DDPHI(5500),DDTEMP(5500),ETRORX(5500),EMRORX(5500)
      COMMON ETROR(5500),EMROR(5500)
      COMMON TRAN(15600,5),ZIP(5500,4)
      COMMON FTS(200),FMS(200),DTFS(200),DMFS(200),FTB(200),
     1 FMB(200),DMFB(200),DTFB(200),TTRANS(200),TMRANS(200),
     1 TSIB(200),PSIB(200),TSIBS(200),PSIBS(200)
      COMMON SMLIM(5500),STLIM(5500),TIMEB(20,200)
      COMMON NMAT,NODES,KCYC,MAX,MID,SET,NOSPEC,NOCON,
     1 NVARH,NOSCON,NODBS,NVART,NMOGEN,NMVARG,
     1 NTOGEN,NTVARG
      COMMON SUMTIM,FOR,FORD,DELTS,DELT,DPIMAX,MAXITT,
     1 SPEED,NUTX,NUTSUM,NUTS,NUTMAX,TAU,S1,S2
      COMMON NUTAVG,PHIVRY,TEMVRY,SMALL,SMALT,BIG,KWIT,
```

```
     1  KDATA,KGOOD,MS,MSS,MSF,DELTMX,DELTO,NREG,DELTG
        COMMON RATTG,RATMG,RATT1,RATM1,RATT2, RATM2,TIMAX,DELTSS,
     1  MCYC,KSPEC,IPRINT,TIMEP
        COMMON NIT
        COMMON/SEEY/INPUT1,INPUT2,NOW
C
C
C
        READ(INPUT1,*) MCYC,TIMAX,TAU,DELTO,SMALL,KDATA
        READ(INPUT1,*) TIMEP,KSPEC,PHIVRY,MAXITT,IPRINT
        WRITE(INPUT2,3)
            TEMVRY=5.0
        WRITE(INPUT2,1)MCYC,TIMAX,TAU,DELTO,SMALL,KDATA
        WRITE(INPUT2,4)
        WRITE(INPUT2,2)TIMEP,KSPEC,PHIVRY,MAXITT,IPRINT
        RETURN
C
2       FORMAT(F12.1,3X,I5,3X,F12.1,2X,F12.1,3X,I5,5X,I5)
1       FORMAT(3X,I5,2X,F12.1,3F12.1,2X,I5)
3       FORMAT(//,5X,4HMCYC,5X,5HTIMAX,5X,3HTAU,5X,5HDELTO,5X
     1  ,5HSMALL,5X,5HKDATA)
4       FORMAT(/,5X,5HTIMEP,5X,5HKSPEC,5X,6HPHIVRY
     1  ,5X,6HMAXITT,5X,6HIPRINT)
        END
C
C
C
C
C
C
C
        SUBROUTINE LINE
        common/SEEY/INPUT1,INPUT2,NOW
        WRITE(INPUT2,100)
        RETURN
100     FORMAT(2X,12(10H==========))
        END
C
C
         SUBROUTINE MATINP
C
C
        COMMON MATL(6),MATAB(6,10),AMX(10,8,6),
     1  AMY(10,8,6),SLOMAT(10,8,6),KCON(6,3)
        COMMON ATCAL(5500,8),NODMAT(5500),NOXMAT(5500),TEMP(5500),
     1  PHI(5500),DTEMP(5500),DPHI(5500),DTF(5500),DMF(5500)
        COMMON VOL(5500),Z(5500),NTYPE(5500),NODE(5500)
        COMMON NOTE(5500),NOXE(5500)
        COMMON NOD1(15600),NOD2(15600),NOX1(15600),NOX2(15600),
     1  DEL1(15600),DEL2(15600),AREA(15600),KODCON(15600)
        COMMON NODS(200),NODSB(200),NOXS(200),NOXSB(200),
     1  HTSURT(20,200),HMSURT(20,200),SMLOH(20,200),STLOH(20,200),
     1  HTSURE(200),HMSURE(200),LTABH(200),AREAS(200),
     1  TMVARH(20,200),TTVARH(20,200)
        COMMON PHIB(20,200),TEMPB(20,200),NODB(200),ZB(200),
     1  LTABT(200),STLOT(20,200),SMLOT(20,200)
        COMMON GMT(20,200),GM(5500),NMOXG(200),NMODG(200),
     1  TMVARG(20,200),LMTABG(200),SMLOG(20,200)
        COMMON GTT(20,200),GT(5500),NTOXG(200),NTODG(200),
     1  TTVARG(20,200),LTTABG(200),STLOG(20,200),
     1  DDPHI(5500),DDTEMP(5500),ETRORX(5500),EMRORX(5500)
        COMMON ETROR(5500),EMROR(5500)
        COMMON TRAN(15600,5),ZIP(5500,4)
        COMMON FTS(200),FMS(200),DTFS(200),DMFS(200),FTB(200),
     1  FMB(200),DMFB(200),DTFB(200),TTRANS(200),TMRANS(200),
```

```
     1  TSIB(200),PSIB(200),TSIBS(200),PSIBS(200)
        COMMON SMLIM(5500),STLIM(5500),TIMEB(20,200)
        COMMON NMAT,NODES,KCYC,MAX,MID,SET,NOSPEC,NOCON,
     1  NVARH,NOSCON,NODBS,NVART,NMOGEN,NMVARG,
     1  NTOGEN,NTVARG
        COMMON SUMTIM,FOR,FORD,DELTS,DELT,DPIMAX,MAXITT,
     1  SPEED,NUTX,NUTSUM,NUTS,NUTMAX,TAU,S1,S2
        COMMON NUTAVG,PHIVRY,TEMVRY,SMALL,SMALT,BIG,KWIT,
     1  KDATA,KGOOD,MS,MSS,MSF,DELTMX,DELTO,NREG,DELTG
        COMMON RATTG,RATMG,RATT1,RATM1,RATT2, RATM2,TIMAX,DELTSS,
     1  MCYC,KSPEC,IPRINT,TIMEP
        COMMON NIT
        COMMON/SEEY/INPUT1,INPUT2,NOW
         NMAT=1
              AABB='SOIL'
              MABB=1
         M=NMAT
         MATL(NMAT)=MABB
         WRITE(INPUT2,500)
         WRITE(INPUT2,1000) AABB,MATL(NMAT)
         DO 2 J=7,7
         READ (INPUT1,1000) AMNAME, MATAB(M,J)
         WRITE(INPUT2,1000) AMNAME, MATAB(M,J)
         IF (MATAB(M,J).LT.2) GO TO 1
         READ(INPUT1,*)(AMX(K,J,M),AMY(K,J,M),K=1,MATAB(M,J))
         DO 5 K=2,MATAB(M,J)
         SLOMAT(K-1,J,M)=(AMY(K,J,M)-AMY(K-1,J,M))/(AMX(K,J,M)
     1   -AMX(K-1,J,M))
5        CONTINUE
         SLOMAT(MATAB(M,J),J,M)=SLOMAT((MATAB(M,J)-1),J,M)
         WRITE (INPUT2,2000)(AMY(K,J,M),AMX(K,J,M),SLOMAT(K,J,M),     1   K=1,MATAB(M,J))
         GO TO 2
1        CONTINUE
         READ(INPUT1,*) AMY(1,J,M)
         WRITE(INPUT2,2000) AMY(1,J,M)
2        CONTINUE
              KCON(M,1)=1
              KCON(M,2)=1
              KCON(M,3)=1
              MATAB(M,8)=1
              AMY(1,8,M)=1.E-13
              DO 99 J=1,6
              MATAB(M,J)=1
              AMY(1,J,M)=1.E-13
99            CONTINUE
3        CONTINUE
         RETURN
1000      FORMAT(1X,A4,I5)
2000      FORMAT(6E12.5)
4000      FORMAT(3I5)
500       FORMAT(1X,4HNAME,3X,7HMAT.NO.)
600       FORMAT(1X,17HANISOTROPIC RATIO)
         END
C
          SUBROUTINE ELEM
C
C
        COMMON MATL(6),MATAB(6,10),AMX(10,8,6),
     1  AMY(10,8,6),SLOMAT(10,8,6),KCON(6,3)
        COMMON ATCAL(5500,8),NODMAT(5500),NOXMAT(5500),TEMP(5500),
     1  PHI(5500),DTEMP(5500),DPHI(5500),DTF(5500),DMF(5500)
        COMMON VOL(5500),Z(5500),NTYPE(5500),NODE(5500)
        COMMON NOTE(5500),NOXE(5500)
        COMMON NOD1(15600),NOD2(15600),NOX1(15600),NOX2(15600),
     1  DEL1(15600),DEL2(15600),AREA(15600),KODCON(15600)
```

```
      COMMON NODS(200),NODSB(200),NOXS(200),NOXSB(200),
     1 HTSURT(20,200),HMSURT(20,200),SMLOH(20,200),STLOH(20,200),
     1 HTSURE(200),HMSURE(200),LTABH(200),AREAS(200),
     1 TMVARH(20,200),TTVARH(20,200)
      COMMON PHIB(20,200),TEMPB(20,200),NODB(200),ZB(200),
     1 LTABT(200),STLOT(20,200),SMLOT(20,200)
      COMMON GMT(20,200),GM(5500),NMOXG(200),NMODG(200),
     1 TMVARG(20,200),LMTABG(200),SMLOG(20,200)
      COMMON GTT(20,200),GT(5500),NTOXG(200),NTODG(200),
     1 TTVARG(20,200),LTTABG(200),STLOG(20,200),
     1 DDPHI(5500),DDTEMP(5500),ETRORX(5500),EMRORX(5500)
      COMMON ETROR(5500),EMROR(5500)
      COMMON TRAN(15600,5),ZIP(5500,4)
      COMMON FTS(200),FMS(200),DTFS(200),DMFS(200),FTB(200),
     1 FMB(200),DMFB(200),DTFB(200),TTRANS(200),TMRANS(200),
     1 TSIB(200),PSIB(200),TSIBS(200),PSIBS(200)
      COMMON SMLIM(5500),STLIM(5500),TIMEB(20,200)
      COMMON NMAT,NODES,KCYC,MAX,MID,SET,NOSPEC,NOCON,
     1 NVARH,NOSCON,NODBS,NVART,NMOGEN,NMVARG,
     1 NTOGEN,NTVARG
      COMMON SUMTIM,FOR,FORD,DELTS,DELT,DPIMAX,MAXITT,
     1 SPEED,NUTX,NUTSUM,NUTS,NUTMAX,TAU,S1,S2
      COMMON NUTAVG,PHIVRY,TEMVRY,SMALL,SMALT,BIG,KWIT,
     1 KDATA,KGOOD,MS,MSS,MSF,DELTMX,DELTO,NREG,DELTG
      COMMON RATTG,RATMG,RATT1,RATM1,RATT2, RATM2,TIMAX,DELTSS,
     1 MCYC,KSPEC,IPRINT,TIMEP
      COMMON NIT
      COMMON/SEEY/INPUT1,INPUT2,NOW
C
      L=0
      NODES=0
      NOSPEC=0
      N=NODES
      KODEZ=0
1     READ(INPUT1,*) N1,NSEQ,NADD,N2,KS,DLONG,DWIDE,DRAD,P2,P1
      IF (N1.EQ.0) RETURN
9     NODES=N+1
      N=NODES
      NOXMAT(N)=N2
        NODMAT(N)=N2
      NODE(N)=N1
      NTYPE(N)=0
      IF (KODEZ.EQ.0) GO TO 3
      Z(N)=Z(N-1)+P2
      GO TO 4
3       Z(N)=P1
4     CONTINUE
      VOL(N)=DLONG*DWIDE*DRAD
      IF (VOL(N).NE.0.0) GO TO 5
      VOL(N)=1.E-10
      NTYPE(N)=2
      GO TO 6
5     IF (KS.EQ.0) GO TO 7
      NTYPE(N)=3
6     NOSPEC=NOSPEC+1
7     IF (MOD(L,57).EQ.0) WRITE (INPUT2,200)
      WRITE (INPUT2,300) N1,N,NODMAT(N),NTYPE(N),DLONG
     1 ,DWIDE,DRAD,VOL(N),Z(N)
      L=L+1
      IF (NSEQ.LE.0) GO TO 8
      NSEQ=NSEQ-1
      N1=N1+NADD
      KODEZ=1
      GO TO 9
8     KODEZ=0
```

```
      GO TO 1
200   FORMAT(//,12X,22HNODE INDEX MATL NTYPE,4X,5HDLONG,10X,
     1 5HWIDE,10X,4HDRAD,11X,6HVOLUME,11X,1HZ)
300   FORMAT(10X,4I5,5E15.5)
      END
C
        SUBROUTINE ELECON
C
C
      COMMON MATL(6),MATAB(6,10),AMX(10,8,6),
     1 AMY(10,8,6),SLOMAT(10,8,6),KCON(6,3)
      COMMON ATCAL(5500,8),NODMAT(5500),NOXMAT(5500),TEMP(5500),
     1 PHI(5500),DTEMP(5500),DPHI(5500),DTF(5500),DMF(5500)
      COMMON VOL(5500),Z(5500),NTYPE(5500),NODE(5500)
      COMMON NOTE(5500),NOXE(5500)
      COMMON NOD1(15600),NOD2(15600),NOX1(15600),NOX2(15600),
     1 DEL1(15600),DEL2(15600),AREA(15600),KODCON(15600)
      COMMON NODS(200),NODSB(200),NOXS(200),NOXSB(200),
     1 HTSURT(20,200),HMSURT(20,200),SMLOH(20,200),STLOH(20,200),
     1 HTSURE(200),HMSURE(200),LTABH(200),AREAS(200),
     1 TMVARH(20,200),TTVARH(20,200)
      COMMON PHIB(20,200),TEMPB(20,200),NODB(200),ZB(200),
     1 LTABT(200),STLOT(20,200),SMLOT(20,200)
      COMMON GMT(20,200),GM(5500),NMOXG(200),NMODG(200),
     1 TMVARG(20,200),LMTABG(200),SMLOG(20,200)
      COMMON GTT(20,200),GT(5500),NTOXG(200),NTODG(200),
     1 TTVARG(20,200),LTTABG(200),STLOG(20,200),
     1 DDPHI(5500),DDTEMP(5500),ETRORX(5500),EMRORX(5500)
      COMMON ETROR(5500),EMROR(5500)
      COMMON TRAN(15600,5),ZIP(5500,4)
      COMMON FTS(200),FMS(200),DTFS(200),DMFS(200),FTB(200),
     1 FMB(200),DMFB(200),DTFB(200),TTRANS(200),TMRANS(200),
     1 TSIB(200),PSIB(200),TSIBS(200),PSIBS(200)
      COMMON SMLIM(5500),STLIM(5500),TIMEB(20,200)
      COMMON NMAT,NODES,KCYC,MAX,MID,SET,NOSPEC,NOCON,
     1 NVARH,NOSCON,NODBS,NVART,NMOGEN,NMVARG,
     1 NTOGEN,NTVARG
      COMMON SUMTIM,FOR,FORD,DELTS,DELT,DPIMAX,MAXITT,
     1 SPEED,NUTX,NUTSUM,NUTS,NUTMAX,TAU,S1,S2
      COMMON NUTAVG,PHIVRY,TEMVRY,SMALL,SMALT,BIG,KWIT,
     1 KDATA,KGOOD,MS,MSS,MSF,DELTMX,DELTO,NREG,DELTG
      COMMON RATTG,RATMG,RATT1,RATM1,RATT2, RATM2,TIMAX,DELTSS,
     1 MCYC,KSPEC,IPRINT,TIMEP
     COMMON NIT
      COMMON/SEEY/INPUT1,INPUT2,NOW
C
        L=0
        NOCON=0
        N=NOCON
100     CONTINUE
1       READ(INPUT1,*) N1,N2,NSEQ,NAD1,NAD2,NZ,P1,P2,DLONG,DRAD
      IF (N1.EQ.0) RETURN
        IF (NZ.LE.0) NZ=1
 2      CONTINUE
        NOCON=N+1
        N=N+1
        KODCON(N)=NZ
        NOD1(N)=N1
        NOX1(N)=N1
        NOD2(N)=N2
        NOX2(N)=N2
        DEL1(N)=P1
        DEL2(N)=P2
        AREA(N)=DLONG*DRAD
        IF (MOD(L,57).EQ.0) WRITE (INPUT2,200)
```

```
      WRITE(INPUT2,300) N1,N2,N,NZ,P1,P2,DLONG,DRAD,AREA(N)
      L=L+1
    IF (NSEQ.LE.0) GO TO 1
      NSEQ=NSEQ-1
      N1=N1+NAD1
      N2=N2+NAD2
      GO TO 2
200   FORMAT(//,11X,'NOD1 NOD2 INDEX KODCON    DEL1',
     1   6X,'DEL2       LONG       DRAD       AREA')
300   FORMAT(11X,4I5,5F10.3)
      END
C
      SUBROUTINE SBCON
C
C
      COMMON MATL(6),MATAB(6,10),AMX(10,8,6),
     1 AMY(10,8,6),SLOMAT(10,8,6),KCON(6,3)
      COMMON ATCAL(5500,8),NODMAT(5500),NOXMAT(5500),TEMP(5500),
     1 PHI(5500),DTEMP(5500),DPHI(5500),DTF(5500),DMF(5500)
      COMMON VOL(5500),Z(5500),NTYPE(5500),NODE(5500)
      COMMON NOTE(5500),NOXE(5500)
      COMMON NOD1(15600),NOD2(15600),NOX1(15600),NOX2(15600),
     1 DEL1(15600),DEL2(15600),AREA(15600),KODCON(15600)
      COMMON NODS(200),NODSB(200),NOXS(200),NOXSB(200),
     1 HTSURT(20,200),HMSURT(20,200),SMLOH(20,200),STLOH(20,200),
     1 HTSURE(200),HMSURE(200),LTABH(200),AREAS(200),
     1 TMVARH(20,200),TTVARH(20,200)
      COMMON PHIB(20,200),TEMPB(20,200),NODB(200),ZB(200),
     1 LTABT(200),STLOT(20,200),SMLOT(20,200)
      COMMON GMT(20,200),GM(5500),NMOXG(200),NMODG(200),
     1 TMVARG(20,200),LMTABG(200),SMLOG(20,200)
      COMMON GTT(20,200),GT(5500),NTOXG(200),NTODG(200),
     1 TTVARG(20,200),LTTABG(200),STLOG(20,200),
     1 DDPHI(5500),DDTEMP(5500),ETRORX(5500),EMRORX(5500)
      COMMON ETROR(5500),EMROR(5500)
      COMMON TRAN(15600,5),ZIP(5500,4)
      COMMON FTS(200),FMS(200),DTFS(200),DMFS(200),FTB(200),
     1 FMB(200),DMFB(200),DTFB(200),TTRANS(200),TMRANS(200),
     1 TSIB(200),PSIB(200),TSIBS(200),PSIBS(200)
      COMMON SMLIM(5500),STLIM(5500),TIMEB(20,200)
      COMMON NMAT,NODES,KCYC,MAX,MID,SET,NOSPEC,NOCON,
     1 NVARH,NOSCON,NODBS,NVART,NMOGEN,NMVARG,
     1 NTOGEN,NTVARG
      COMMON SUMTIM,FOR,FORD,DELTS,DELT,DPIMAX,MAXITT,
     1 SPEED,NUTX,NUTSUM,NUTS,NUTMAX,TAU,S1,S2
      COMMON NUTAVG,PHIVRY,TEMVRY,SMALL,SMALT,BIG,KWIT,
     1 KDATA,KGOOD,MS,MSS,MSF,DELTMX,DELTO,NREG,DELTG
      COMMON RATTG,RATMG,RATT1,RATM1,RATT2, RATM2,TIMAX,DELTSS,
     1 MCYC,KSPEC,IPRINT,TIMEP
    COMMON NIT
      COMMON/SEEY/INPUT1,INPUT2,NOW
C
C
      DIMENSION FT1(20),FT2(20),FT3(20),FM1(20),FM2(20),FM3(20)
      L=0
      LABEL=0
      NVARH=0
      N=0
      NOSCON=0
10    READ(INPUT1,*) N1,N2,NSEQ,NADS,NADSB,L1,DLONG,DRAD,
     1 PTATNX,PMATNX
    IF (N1.EQ.0) RETURN
      LTAB=IABS(L1)
      IF (LTAB.GT.1) GO TO 12
      L1=0
```

```
      LTAB=0
      GO TO 15
12    NVARH=NVARH+1+NSEQ
25    READ (INPUT1,*)(FT1(J),FT2(J),J=1,LTAB)
      READ (INPUT1,*)(FM1(J),FM2(J),J=1,LTAB)
      DO 1 J=2,LTAB
      FT3(J-1)=(FT1(J)-FT1(J-1))/(FT2(J)-FT2(J-1))
      FM3(J-1)=(FM1(J)-FM1(J-1))/(FM2(J)-FM2(J-1))
1     CONTINUE
      FT3(LTAB)=FT3(LTAB-1)
      FM3(LTAB)=FM3(LTAB-1)
15    NOSCON=N+1
      N=N+1
      NODS(N)=N1
      NOXS(N)=N1
      NODSB(N)=N2
      NOXSB(N)=N2
      HTSURT(1,N)=PTATNX
      HMSURT(1,N)=PMATNX
      HMSURE(N)=PMATNX
      HTSURE(N)=PTATNX
      LTABH(N)=L1
      AREAS(N)=DLONG*DRAD
      IF (LABEL.EQ.0) GO TO 17
      LABEL=0
      GO TO 75
17    IF (MOD(L,57).NE.0) GO TO 18
75    WRITE (INPUT2,300)
18    WRITE (INPUT2,400) N1,N2,N,L1,DLONG,DRAD,PTATNX,
     1 PMATNX,AREAS(N)
      L=L+1
      IF (LTAB.EQ.0) GO TO 19
      DO 85 J=1,LTAB
      HTSURT(J,N)=FT1(J)
      TTVARH(J,N)=FT2(J)
      STLOH(J,N)=FT3(J)
      HMSURT(J,N)=FM1(J)
      TMVARH(J,N)=FM2(J)
      SMLOH(J,N)=FM3(J)
85    CONTINUE
      LABEL=1
      WRITE (INPUT2,500)
      WRITE (INPUT2,600)(HTSURT(J,N),STLOH(J,N),TTVARH(J,N),
     1 J=1,LTAB)
      WRITE (INPUT2,700)
      WRITE (INPUT2,600)(HMSURT(J,N),SMLOH(J,N),TMVARH(J,N),
     1    J=1,LTAB)
19    IF (NSEQ.LE.0) GO TO 10
      NSEQ=NSEQ-1
      N1=N1+NADS
      N2=N2+NADSB
      GO TO 15
300   FORMAT(/,10X,'NODSB NODS   INDEX   LTABH      DLONG      DRAD,'
     1      5X,'HTSURE      HMSURE      AREAS')
400   FORMAT(8X,4I6,2X,5F10.3)
500   FORMAT(/,5X,'TEMP  SLOPE   TIME')
600   FORMAT(1X,3E12.5)
700   FORMAT(/,5X,'MOISTURE SLOPE TIME')
      END
C
      SUBROUTINE SURF
C
C
     COMMON MATL(6),MATAB(6,10),AMX(10,8,6),
     1  AMY(10,8,6),SLOMAT(10,8,6),KCON(6,3)
```

```
      COMMON ATCAL(5500,8),NODMAT(5500),NOXMAT(5500),TEMP(5500),
     1 PHI(5500),DTEMP(5500),DPHI(5500),DTP(5500),DMF(5500)
      COMMON VOL(5500),Z(5500),NTYPE(5500),NODE(5500)
      COMMON NOTE(5500),NOXE(5500)
      COMMON NOD1(15600),NOD2(15600),NOX1(15600),NOX2(15600),
     1 DEL1(15600),DEL2(15600),AREA(15600),KODCON(15600)
      COMMON NODS(200),NODSB(200),NOXS(200),NOXSB(200),
     1 HTSURT(20,200),HMSURT(20,200),SMLOH(20,200),STLOH(20,200),
     1 HTSURE(200),HMSURE(200),LTABH(200),AREAS(200),
     1 TMVARH(20,200),TTVARH(20,200)
      COMMON PHIB(20,200),TEMPB(20,200),NODB(200),ZB(200),
     1 LTABT(200),STLOT(20,200),SMLOT(20,200)
      COMMON GMT(20,200),GM(5500),NMOXG(200),NMODG(200),
     1 TMVARG(20,200),LMTABG(200),SMLOG(20,200)
      COMMON GTT(20,200),GT(5500),NTOXG(200),NTODG(200),
     1 TTVARG(20,200),LTTABG(200),STLOG(20,200),
     1 DDPHI(5500),DDTEMP(5500),ETRORX(5500),EMRORX(5500)
      COMMON ETROR(5500),EMROR(5500)
      COMMON TRAN(15600,5),ZIP(5500,4)
      COMMON FTS(200),FMS(200),DTFS(200),DMFS(200),FTB(200),
     1 FMB(200),DMFB(200),DTFB(200),TTRANS(200),TMRANS(200),
     1 TSIB(200),PSIB(200),TSIBS(200),PSIBS(200)
      COMMON SMLIM(5500),STLIM(5500),TIMEB(20,200)
      COMMON NMAT,NODES,KCYC,MAX,MID,SET,NOSPEC,NOCON,
     1 NVARH,NOSCON,NODBS,NVART,NMOGEN,NMVARG,
     1 NTOGEN,NTVARG
      COMMON SUMTIM,FOR,FORD,DELTS,DELT,DPIMAX,MAXITT,
     1 SPEED,NUTX,NUTSUM,NUTS,NUTMAX,TAU,S1,S2
      COMMON NUTAVG,PHIVRY,TEMVRY,SMALL,SMALT,BIG,KWIT,
     1 KDATA,KGOOD,MS,MSS,MSF,DELTMX,DELTO,NREG,DELTG
      COMMON RATTG,RATMG,RATT1,RATM1,RATT2, RATM2,TIMAX,DELTSS,
     1 MCYC,KSPEC,IPRINT,TIMEP
      COMMON NIT
      COMMON/SEEY/INPUT1,INPUT2,NOW
C
C
      DIMENSION FT1(20),FM1(20),F2(20)
      LABEL=1
      NODBS=0
      NVART=0
      N=NODBS
5     READ (INPUT1,*) N1,L1,P1,(FT1(J),FM1(J),F2(J),J=1,3)
      IF (N1.EQ.0) RETURN
21    NODBS=N+1
      N=N+1
      DO 23 J=1,3
      PHIB(J,N)=FM1(J)
      TEMPB(J,N)=FT1(J)
23    TIMEB(J,N)=F2(J)
      NODB(N)=N1
      ZB(N)=P1
      PSIBS(N)=FM1(1)
      PSIB(N)=FM1(1)
      TSIBS(N)=FT1(1)
      TSIB(N)=FT1(1)
      LTAB=IABS(L1)
      LTABT(N)=LTAB
      IF (LTAB.LT.100) GO TO 35
      LTAB=2
      STLOT(2,N)=0.0
      SMLOT(2,N)=0.0
      STLOT(1,N)=0.0
      SMLOT(1,N)=0.0
      IF (TIMEB(1,N).LE.0.0) TIMEB(1,N)=1.E24
35    IF (LABEL.LE.0) GO TO 24
```

```
      LABEL=0
      WRITE (INPUT2,200)
24    WRITE (INPUT2,300) N1,N,LTAB,ZB(N),FT1(1),FM1(1),TIMEB(1,N)
      IF (LTAB.GT.1) GO TO 45
      LTABT(N)=0
      GO TO 5
45    NVART=NVART+1
      IF (LTAB.LT.4) GO TO 25
      LABEL=1
      READ (INPUT1,*)(TEMPB(J,N),PHIB(J,N),TIMEB(J,N),J=4,LTAB)
25    IF (LTABT(N).GE.100) GO TO 56
      DO 1 J=2,LTAB
      STLOT(J-1,N)=(TEMPB(J,N)-TEMPB(J-1,N))/(TIMEB(J,N)-
     1 TIMEB(J-1,N))
      SMLOT(J-1,N)=(PHIB(J,N)-PHIB(J-1,N))/(TIMEB(J,N)-
     1 TIMEB(J-1,N))
1     CONTINUE
      STLOT(LTAB,N)=STLOT(LTAB-1,N)
      SMLOT(LTAB,N)=SMLOT(LTAB-1,N)
      WRITE(INPUT2,500)(TEMPB(J,N),STLOT(J,N),TIMEB(J,N),J=1,LTAB)
      WRITE(INPUT2,500)(PHIB(J,N),SMLOT(J,N),TIMEB(J,N),J=1,LTAB)
56    CONTINUE
      IF (LTABT(N).GE.100) WRITE (INPUT2,600)
      GO TO 5
200   FORMAT(//,12X,'NODB INDEX  LTABT     ZB       H        M',
     1   9X,'TIME')
300   FORMAT (10X,3I6,3F10.3,1F10.3)
500   FORMAT (28X,3F10.3)
600   FORMAT(//,8X,'TB=TEMPB(1)+TEMB(2)*SIN(2PI(SUMTIM+TIMEB(2)
     1 /TIMEB(1)')
      END
C
      SUBROUTINE MGEN
C
C
C
      COMMON MATL(6),MATAB(6,10),AMX(10,8,6),
     1 AMY(10,8,6),SLOMAT(10,8,6),KCON(6,3)
      COMMON ATCAL(5500,8),NODMAT(5500),NOXMAT(5500),TEMP(5500),
     1 PHI(5500),DTEMP(5500),DPHI(5500),DTF(5500),DMF(5500)
      COMMON VOL(5500),Z(5500),NTYPE(5500),NODE(5500)
      COMMON NOTE(5500),NOXE(5500)
      COMMON NOD1(15600),NOD2(15600),NOX1(15600),NOX2(15600),
     1 DEL1(15600),DEL2(15600),AREA(15600),KODCON(15600)
      COMMON NODS(200),NODSB(200),NOXS(200),NOXSB(200),
     1 HTSURT(20,200),HMSURT(20,200),SMLOH(20,200),STLOH(20,200),
     1 HTSURE(200),HMSURE(200),LTABH(200),AREAS(200),
     1 TMVARH(20,200),TTVARH(20,200)
      COMMON PHIB(20,200),TEMPB(20,200),NODB(200),ZB(200),
     1 LTABT(200),STLOT(20,200),SMLOT(20,200)
      COMMON GMT(20,200),GM(5500),NMOXG(200),NMODG(200),
     1 TMVARG(20,200),LMTABG(200),SMLOG(20,200)
      COMMON GTT(20,200),GT(5500),NTOXG(200),NTODG(200),
     1 TTVARG(20,200),LTTABG(200),STLOG(20,200),
     1 DDPHI(5500),DDTEMP(5500),ETRORX(5500),EMRORX(5500)
      COMMON ETROR(5500),EMROR(5500)
      COMMON TRAN(15600,5),ZIP(5500,4)
      COMMON FTS(200),FMS(200),DTFS(200),DMFS(200),FTB(200),
     1 FMB(200),DMFB(200),DTFB(200),TTRANS(200),TMRANS(200),
     1 TSIB(200),PSIB(200),TSIBS(200),PSIBS(200)
      COMMON SMLIM(5500),STLIM(5500),TIMEB(20,200)
      COMMON NMAT,NODES,KCYC,MAX,MID,SET,NOSPEC,NOCON,
     1 NVARH,NOSCON,NODBS,NVART,NMOGEN,NMVARG,
     1 NTOGEN,NTVARG
      COMMON SUMTIM,FOR,FORD,DELTS,DELT,DPIMAX,MAXGTT,
```

```
     1  SPEED,NUTX,NUTSUM,NUTS,NUTMAX,TAU,S1,S2
        COMMON NUTAVG,PHIVRY,TEMVRY,SMALL,SMALT,BIG,KWIT,
     1  KDATA,KGOOD,MS,MSS,MSP,DELTMX,DELTO,NREG,DELTO
        COMMON RATTO,RATM0,RATT1,RATM1,RATT2, RATM2,TIMAX,DELTSS,
     1  MCYC,KSPEC,IPRINT,TIMEP
        COMMON /NT
        COMMON/SEEY/INPUT1,INPUT2,NOW
          DIMENSION F1(20),F2(20),F3(20)
          LABEL=1
          NMOGEN=0
          NMVARG=0
          N=0
10        READ(INPUT1,*) N1,NSEQ,NADG,L1,(F1(J),F2(J),J=1,3)
          IF (N1.EQ.0) RETURN
          LTAB=IABS(L1)
          IF (LTAB.NE.0) GO TO 20
          IF (F2(1).EQ.0) F2(1)=1.E-24
          GO TO 30
20        IF (LTAB.LT.4) GO TO 40
30        READ (INPUT1,*)(F1(J),F2(J),J=4,LTAB)
40        CONTINUE
          IF (LTAB.LE.1) GO TO 30
          DO 1 J=2,LTAB
          F3(J-1)=(F1(J)-F1(J-1))/(F2(J)-F2(J-1))
1         CONTINUE
          F3(LTAB)=F3(LTAB-1)
50        NMOGEN=N+1
          N=N+1
55        NMOX(G(N)=N1
          NMODG(N)=N1
          GMT(1,N)=F1(1)
          TMVARG(1,N)=F2(1)
          LMTABG(N)=L1
          IF (L1.NE.-1) GO TO 60
          WRITE (INPUT2,300)
          GO TO 65
60        IF (LABEL.EQ.0) GO TO 95
65        LABEL=0
          WRITE (INPUT2,600)
95        WRITE (INPUT2,400) N1,N,L1,F1(1),F2(1)
          IF (LTAB.LE.1) GO TO 90
          NMVARG=NMVARG+1
          DO 85 J=1,LTAB
          GMT(J,N)=F1(J)
          TMVARG(J,N)=F2(J)
85        CONTINUE
          IF (LTAB.EQ.-1) GO TO 90
          DO 80 J=2,LTAB
          SMLOG(J-1,N)=(F1(J)-F1(J-1))/(F2(J)-F2(J-1))
80        CONTINUE
          SMLOG(LTAB,N)=SMLOG(LTAB-1,N)
          WRITE (INPUT2,500) (GMT(J,N),SMLOG(J,N),TMVARG(J,N),
     1    J=1,LTAB)
90        IF (NSEQ.LE.0) GO TO 10
          NSEQ=NSEQ-1
          N1=N1+NADG
          GO TO 30
300       FORMAT(/,10X,'FOLLOWING NODE HAS G(N)=GM(1,N)*EXP(
     1    -0.693 *S1/M1M/TMVARG(1,N)')
400       FORMAT(10X,I6A,1P10.3,15X,1P10.3)
500       FORMAT(31X,3P10.3)
600       FORMAT(/,12X,16HNODG INDEX LTABG,8X,2HGM,12X,5HSLOPE,
     1    11X,6HTMVARG)
          END
C
```

```
      SUBROUTINE INICON
C
C
C
      COMMON MATL(8),MATAB(6,10),AMX(10,8,6),
     1 AMY(10,8,6),SLOMAT(10,8,6),KCON(6,3)
      COMMON ATCAL(5500,8),NODMAT(5500),NOXMAT(5500),TEMP(5500),
     1 PHI(5500),DTEMP(5500),DPHI(5500),DTF(5500),DMF(5500)
      COMMON VOL(5500),Z(5500),NTYPE(5500),NODE(5500)
      COMMON NOTE(5500),NOXE(5500)
      COMMON NOD1(15600),NOD2(15600),NOX1(15600),NOX2(15600),
     1 DEL1(15600),DEL2(15600),AREA(15600),KODCON(15600)
      COMMON NODS(200),NODSB(200),NOXS(200),NOXSB(200),
     1 HTSURT(20,200),HMSURT(20,200),SMLOH(20,200),STLOH(20,200),
     1 HTSURB(200),HMSURB(200),LTABH(200),AREAS(200),
     1 TMVARH(20,200),TTVARH(20,200)
      COMMON PHIB(20,200),TEMPB(20,200),NODB(200),ZB(200),
     1 LTABT(200),STLOT(20,200),SMLOT(20,200)
      COMMON GMT(20,200),GM(5500),NMOXG(200),NMODG(200),
     1 TMVARG(20,200),LMTABG(200),SMLOG(20,200)
      COMMON GTT(20,200),GT(5500),NTOXG(200),NTODG(200),
     1 TTVARG(20,200),LTTABG(200),STLOG(20,200),
     1 DDPHI(5500),DDTEMP(5500),ETRORX(5500),EMRORX(5500)
      COMMON ETROR(5500),EMROR(5500)
      COMMON TRAN(15600,5),ZIP(5500,4)
      COMMON FTS(200),FMS(200),DTFS(200),DMFS(200),FTB(200),
     1 FMB(200),DMFB(200),DTFB(200),TTRANS(200),TMRANS(200),
     1 TSIBS(200),PSIBS(200),TSIBS(200),PSIBS(200)
      COMMON SM1,IM(5500),STLIM(5500),TIMEB(20,200)
      COMMON NMAT,NODES,KCYC,MAX,MID,SET,NOSPEC,NOCON,
     1 NVARH,NOBCON,NODES,NVART,NMOGEN,NMVARG,
     1 NTOGEN,NTVARG
      COMMON SUMTIM,FOR,FORD,DELTS,DELT,DPIMAX,MAXITT,
     1 SPEED,NUTX,NUTSUM,NUTS,NUTMAX,TAU,S1,S2
      COMMON NUTAVG,PHIVRY,TEMVRY,SMALL,SMALT,BIG,KWIT,
     1 KDATA,IGOOD,MS,MSS,MSF,DELTMX,DELTO,NREG,DELTG
      COMMON RATTG,RATMG,RATT1,RATM1,RATT2, RATM2,TIMAX,DELTS5,
     1 MCYC,KSPEC,IPRINT,TIMEP
      COMMON NT
      COMMON/SEEY/INPUT1,INPUT2,NOW

C
      L=0
      NIT=0
      N=NIT
30    READ(INPUT1,*) N1,NSEQ,NADD,PMATN1,PMATN2
      IF (N1.EQ.0) RETURN
31    NIT=N+1
      N=N+1
      NOXE(N)=N1
      NOTE(N)=N1
       DDPHI(N)=PMATN1
       DDTEMP(N)=0.0
       ETROR(N)=0.0
       EMROR(N)=PMATN2
      IF (MOD(L,57).EQ.0) WRITE (INPUT2,100)
      WRITE (INPUT2,200) N1,N,PMATN1,PMATN2
      L=L+1
      IF (NSEQ.LE.0) GO TO 30
      NSEQ=NSEQ-1
      N1=N1+NADD
      GO TO 31
100   FORMAT (//,12X,'NODE INDEX       PHI       MGEN     ')
200   FORMAT (10X,2I5,4X,4F10.3)
      END
```

```
C
      SUBROUTINE INTST
C
C
C
      COMMON MATL(6),MATAB(6,10),AMX(10,8,6),
     1 AMY(10,8,6),SLOMAT(10,8,6),KCON(6,3)
      COMMON ATCAL(5500,8),NODMAT(5500),NOXMAT(5500),TEMP(5500),
     1 PHI(5500),DTEMP(5500),DPHI(5500),DTF(5500),DMF(5500)
      COMMON VOL(5500),Z(5500),NTYPE(5500),NODE(5500)
      COMMON NOTE(5500),NOXE(5500)
      COMMON NOD1(15600),NOD2(15600),NOX1(15600),NOX2(15600),
     1 DEL1(15600),DEL2(15600),AREA(15600),KODCON(15600)
      COMMON NODS(200),NODSB(200),NOXS(200),NOXSB(200),
     1 HTSURT(20,200),HMSURT(20,200),SMLOH(20,200),STLOH(20,200),
     1 HTSURE(200),HMSURE(200),LTABH(200),AREAS(200),
     1 TMVARH(20,200),TTVARH(20,200)
      COMMON PHIB(20,200),TEMPB(20,200),NODB(200),ZB(200),
     1 LTABT(200),STLOT(20,200),SMLOT(20,200)
      COMMON GMT(20,200),GM(5500),NMOXG(200),NMODG(200),
     1 TMVARG(20,200),LMTABG(200),SMLOG(20,200)
      COMMON GTT(20,200),GT(5500),NTOXG(200),NTODG(200),
     1 TTVARG(20,200),LTTABG(200),STLOG(20,200),
     1 DDPHI(5500),DDTEMP(5500),ETRORX(5500),EMRORX(5500)
      COMMON ETROR(5500),EMROR(5500)
      COMMON TRAN(15600,5),ZIP(5500,4)
      COMMON FTS(200),FMS(200),DTFS(200),DMFS(200),FTB(200),
     1 FMB(200),DMFB(200),DTFB(200),TTRANS(200),TMRANS(200),
     1 TSIB(200),PSIB(200),TSIBS(200),PSIBS(200)
      COMMON SMLIM(5500),STLIM(5500),TIMEB(20,200)
      COMMON NMAT,NODES,KCYC,MAX,MID,SET,NOSPEC,NOCON,
     1 NVARH,NOSCON,NODBS,NVART,NMOGEN,NMVARG,
     1 NTOGEN,NTVARG
      COMMON SUMTIM,FOR,FORD,DELTS,DELT,DPIMAX,MAXITT,
     1 SPEED,NUTX,NUTSUM,NUTS,NUTMAX,TAU,S1,S2
      COMMON NUTAVG,PHIVRY,TEMVRY,SMALL,SMALT,BIG,KWIT,
     1 KDATA,KGOOD,MS,MSS,MSF,DELTMX,DELTO,NREG,DELTG
      COMMON RATTG,RATMG,RATT1,RATM1,RATT2, RATM2,TIMAX,DELTSS,
     1 MCYC,KSPEC,IPRINT,TIMEP
     COMMON NIT
      COMMON/SEEY/INPUT1,INPUT2,NOW
C
C
C
      CALL REFER(NOTE,NOXE,NIT,NODE,NODES)
      DO 1 N=1,NIT
      NND=NOTE(N)
       GM(NND)=EMRORX(N)
      GT(NND)=ETRORX(N)
      PHI(NND)=DDPHI(N)
      TEMP(NND)=DDTEMP(N)
1     CONTINUE
      WRITE (INPUT2,500) NMAT,NODES
      IF (KSPEC.LE.0) GO TO 50
      DO 35 N=1,NODES
      IF (NTYPE(N).NE.0) GO TO 35
      NTYPE(N)=5
      NOSPEC=NOSPEC+1
35    CONTINUE
50    NTOGEN=0
      NMOGEN=0
      DO 65 N=1,NODES
      IF (GT(N).EQ.0) GO TO 55
      NTOGEN=NTOGEN+1
```

```
55      IF (GM(N).EQ.0) GO TO 65
        NMOGEN=NMOGEN+1
65      CONTINUE
        NTOGEN=NTOGEN+NTVARG
        NMOGEN=NMOGEN+NMVARG
        SUMTIM=TAU
        FOR=1.0
        DELT=1.0E-12
        FORD=FOR*DELT
        DPIMAX=0.0
        DELTS=0.0
        DO 82 N=1,NODES
        DDPHI(N)=0.0
        DDTEMP(N)=0.0
82      CONTINUE
        NUTX=0
        NUTSUM=0
        NUTS=0
        NUTMAX=MAXITT
        DO 100 N=1,NODES
        ETRORX(N)=0.0
100     EMRORX(N)=0.0
        SPEED=0.2
        RETURN
500     FORMAT(1X,'MAT=    ',I3,'   NODES=',I4)
        END
C
        SUBROUTINE MATCAL
C
C
C
      COMMON MATL(6),MATAB(6,10),AMX(10,8,6),
     1 AMY(10,8,6),SLOMAT(10,8,6),KCON(6,3)
      COMMON ATCAL(5500,8),NODMAT(5500),NOXMAT(5500),TEMP(5500),
     1 PHI(5500),DTEMP(5500),DPHI(5500),DTF(5500),DMF(5500)
      COMMON VOL(5500),Z(5500),NTYPE(5500),NODE(5500)
      COMMON NOTE(5500),NOXE(5500)
      COMMON NOD1(15600),NOD2(15600),NOX1(15600),NOX2(15600),
     1 DEL1(15600),DEL2(15600),AREA(15600),KODCON(15600)
      COMMON NODS(200),NODSB(200),NOXS(200),NOXSB(200),
     1 HTSURT(20,200),HMSURT(20,200),SMLOH(20,200),STLOH(20,200),
     1 HTSURE(200),HMSURE(200),LTABH(200),AREAS(200),
     1 TMVARH(20,200),TTVARH(20,200)
      COMMON PHIB(20,200),TEMPB(20,200),NODB(200),ZB(200),
     1 LTABT(200),STLOT(20,200),SMLOT(20,200)
      COMMON GMT(20,200),GM(5500),NMOXG(200),NMODG(200),
     1 TMVARG(20,200),LMTABG(200),SMLOG(20,200)
      COMMON GTT(20,200),GT(5500),NTOXG(200),NTODG(200),
     1 TTVARG(20,200),LTTABG(200),STLOG(20,200),
     1 DDPHI(5500),DDTEMP(5500),ETRORX(5500),EMRORX(5500)
      COMMON ETROR(5500),EMROR(5500)
      COMMON TRAN(15600,5),ZIP(5500,4)
      COMMON PTS(200),PMS(200),DTFS(200),DMFS(200),PTB(200),
     1 PMB(200),DMFB(200),DTFB(200),TTRANS(200),TMRANS(200),
     1 TSIB(200),PSIB(200),TSIBS(200),PSIBS(200)
      COMMON SMLIM(5500),STLIM(5500),TIMEB(20,200)
      COMMON NMAT,NODES,KCYC,MAX,MID,SET,NOSPEC,NOCON,
     1 NVARH,NOSCON,NODBS,NVART,NMOGEN,NMVARG,
     1 NTOGEN,NTVARG
      COMMON SUMTIM,FOR,FORD,DELTS,DELT,DPIMAX,MAXITT,
     1 SPEED,NUTX,NUTSUM,NUTS,NUTMAX,TAU,S1,S2
      COMMON NUTAVG,PHIVRY,TEMVRY,SMALL,SMALT,BIG,KWIT,
     1 KDATA,KGOOD,MS,MSS,MSF,DELTMX,DELTO,NREG,DELTG
      COMMON RATTG,RATMG,RATT1,RATM1,RATT2, RATM2,TIMAX,DELTSS,
     1 MCYC,KSPEC,IPRINT,TIMEP
```

```
      COMMON NIT
      COMMON/SEEY/INPUT1,INPUT2,NOW
c
c
C
      IF (KCYC.GT.0) GO TO 1
      CALL REFER(NODMAT,NOXMAT,NODES,MATL,NMAT)
1     CONTINUE
      DO 8 N=1,NODES
      ZIP(N,4)=0.0
      ZIP(N,3)=0.0
      ZIP(N,2)=0.0
      ZIP(N,1)=0.0
      DPHI(N)=0.0
      DTEMP(N)=0.0
      DTF(N)=0.0
      DMF(N)=0.0
      NND=NODMAT(N)
c     SET=TEMP(N)+DDTEMP(N)*FORD
cDO 4 J=1,5
ccIF (MATAB(NND,J).LE.1) GO TO 3
ccMAX=MATAB(NND,J)
cCALL MIDPT(AMX(1,J,NND))
cATCAL(N,J)=AMY(MID,J,NND)+SLOMAT(MID,J,NND)
c    1            *(SET-AMX(MID,J,NND))
C     WRITE(INPUT2,*)MID,AMX(MID,J,NND),SET,AMY(MID,J,NND)
C    1 ,SLOMAT(MID,J,NND)

cGO TO 4
c3    ATCAL(N,J)=AMY(1,J,NND)
c4    CONTINUE
      SET=PHI(N)+DDPHI(N)*FORD
      DO 6 J=1,8
      IF (MATAB(NND,J).LE.1) GO TO 7
      MAX=MATAB(NND,J)
      CALL MIDPT(AMX(1,J,NND))
      ATCAL(N,J)=AMY(MID,J,NND)+SLOMAT(MID,J,NND)
     1   *(SET-AMX(MID,J,NND))
      ATCAL(N,J)=10**ATCAL(N,J)
c     WRITE(INPUT2,*)MID,AMX(MID,J,NND),SET,AMY(MID,J,NND)
c    1 ,SLOMAT(MID,J,NND)
      GO TO 6
7     ATCAL(N,J)=AMY(1,J,NND)
6     CONTINUE
C     WRITE(INPUT2,1010)
C     WRITE(INPUT2,*)PHI(N),DDPHI(N),TEMP(N),DDTEMP(N),FORD
8     CONTINUE
C1010 FORMAT(1X,'PHI(N),DDPHI(N),TEMP(N),DDTEMP(N),FORD')
C     WRITE(INPUT2,1000)(ATCAL(1,J),J=1,8)
C1000 FORMAT(1X,'ATCAL',8F10.3)
      RETURN
      END
C
      SUBROUTINE SOURCE
C
C
C
      COMMON MATL(6),MATAB(6,10),AMX(10,8,6),
     1 AMY(10,8,6),SLOMAT(10,8,6),KCON(6,3)
      COMMON ATCAL(5500,8),NODMAT(5500),NOXMAT(5500),TEMP(5500),
     1 PHI(5500),DTEMP(5500),DPHI(5500),DTF(5500),DMF(5500)
      COMMON VOL(5500),Z(5500),NTYPE(5500),NODE(5500)
      COMMON NOTE(5500),NOXE(5500)
      COMMON NOD1(15600),NOD2(15600),NOX1(15600),NOX2(15600),
     1 DEL1(15600),DEL2(15600),AREA(15600),KODCON(15600)
```

```
      COMMON NODS(200),NODSB(200),NOXS(200),NOXSB(200),
     1 HTSURT(20,200),HMSURT(20,200),SMLOH(20,200),STLOH(20,200),
     1 HTSURE(200),HMSURE(200),LTABH(200),AREAS(200),
     1 TMVARH(20,200),TTVARH(20,200)
      COMMON PHIB(20,200),TEMPB(20,200),NODB(200),ZB(200),
     1 LTABT(200),STLOT(20,200),SMLOT(20,200)
      COMMON GMT(20,200),GM(5500),NMOXG(200),NMODG(200),
     1 TMVARG(20,200),LMTABG(200),SMLOG(20,200)
      COMMON GTT(20,200),GT(5500),NTOXG(200),NTODG(200),
     1 TTVARG(20,200),LTTABG(200),STLOG(20,200),
     1 DDPHI(5500),DDTEMP(5500),ETRORX(5500),EMRORX(5500)
      COMMON ETROR(5500),EMROR(5500)
      COMMON TRAN(15600,5),ZIP(5500,4)
      COMMON FTS(200),FMS(200),DTFS(200),DMFS(200),FTB(200),
     1 FMB(200),DMFB(200),DTFB(200),TTRANS(200),TMRANS(200),
     1 TSIB(200),PSIB(200),TSIBS(200),PSIBS(200)
      COMMON SMLIM(5500),STLIM(5500),TIMEB(20,200)
      COMMON NMAT,NODES,KCYC,MAX,MID,SET,NOSPEC,NOCON,
     1 NVARH,NOSCON,NODBS,NVART,NMOGEN,NMVARG,
     1 NTOGEN,NTVARG
      COMMON SUMTIM,FOR,FORD,DELTS,DELT,DPIMAX,MAXITT,
     1 SPEED,NUTX,NUTSUM,NUTS,NUTMAX,TAU,S1,S2
      COMMON NUTAVG,PHIVRY,TEMVRY,SMALL,SMALT,BIG,KWIT,
     1 KDATA,KGOOD,MS,MSS,MSF,DELTMX,DELTO,NREG,DELTG
      COMMON RATTG,RATMG,RATT1,RATM1,RATT2, RATM2,TIMAX,DELTSS,
     1 MCYC,KSPEC,IPRINT,TIMEP
      COMMON NIT
      COMMON/SEEY/INPUT1,INPUT2,NOW
C
C
C
      IF (KCYC.GT.0) GO TO 10
      IF (NTVARG.LE.0) GO TO 5
      CALL REFER(NTODG,NTOXG,NTOGEN,NODE,NODES)
      CALL REFER(NMODG,NMOXG,NMOGEN,NODE,NODES)
5     IF (KWIT.NE.0) RETURN
10    IF (NTVARG.LE.0) GO TO 30
      DO 40 N=1,NTVARG
      NND=NTODG(N)
      MAX=LTTABG(N)
      IF (MAX) 50,60,60
50    MAX=-MAX
      SET=SUMTIM+FORD
      GO TO 70
60    SET=TEMP(NND)+DDTEMP(NND)*FORD
70    CONTINUE
      CALL MIDPT(TTVARG(1,N))
      GT(NND)=GTT(MID,N)+STLOG(MID,N)*(SET-TTVARG(MID,N))
40    CONTINUE
30    DO 80 N=1,NODES
      DTEMP(N)=DTEMP(N)+GT(N)*DELT/VOL(N)/ATCAL(N,1)
80    CONTINUE
      IF (NMVARG.LE.0) GO TO 130
      DO 90 N=1,NMVARG
      NND=NMODG(N)
      MAX=LMTABG(N)
      IF (MAX) 100,110,110
100   MAX=-MAX
      SET=SUMTIM+FORD
      GO TO 120
110   SET=PHI(NND)+DDPHI(NND)*FORD
120   CONTINUE
      CALL MIDPT(TMVARG(1,N))
      GM(NND)=GMT(MID,N)+SMLOG(MID,N)*(SET-TMVARG(MID,N))
90    CONTINUE
```

```
130       DO 140 N=1,NODES
          DPHI(N)=DPHI(N)+GM(N)*DELT/VOL(N)
140       CONTINUE
          end
C
          SUBROUTINE ESOLVE
C
C
C
          COMMON MATL(6),MATAB(6,10),AMX(10,8,6),
     1    AMY(10,8,6),SLOMAT(10,8,6),KCON(6,3)
          COMMON ATCAL(5500,8),NODMAT(5500),NOXMAT(5500),TEMP(5500),
     1    PHI(5500),DTEMP(5500),DPHI(5500),DTF(5500),DMF(5500)
          COMMON VOL(5500),Z(5500),NTYPE(5500),NODE(5500)
          COMMON NOTE(5500),NOXE(5500)
          COMMON NOD1(15600),NOD2(15600),NOX1(15600),NOX2(15600),
     1    DEL1(15600),DEL2(15600),AREA(15600),KODCON(15600)
          COMMON NODS(200),NODSB(200),NOXS(200),NOXSB(200),
     1    HTSURT(20,200),HMSURT(20,200),SMLOH(20,200),STLOH(20,200),
     1    HTSURE(200),HMSURE(200),LTABH(200),AREAS(200),
     1    TMVARH(20,200),TTVARH(20,200)
          COMMON PHIB(20,200),TEMPB(20,200),NODB(200),ZB(200),
     1    LTABT(200),STLOT(20,200),SMLOT(20,200)
          COMMON GMT(20,200),GM(5500),NMOXG(200),NMODG(200),
     1    TMVARG(20,200),LMTABG(200),SMLOG(20,200)
          COMMON GTT(20,200),GT(5500),NTOXG(200),NTODG(200),
     1    TTVARG(20,200),LTTABG(200),STLOG(20,200),
     1    DDPHI(5500),DDTEMP(5500),ETRORX(5500),EMRORX(5500)
          COMMON ETROR(5500),EMROR(5500)
          COMMON TRAN(15600,5),ZIP(5500,4)
          COMMON FTS(200),FMS(200),DTFS(200),DMFS(200),FTB(200),
     1    FMB(200),DMFB(200),DTFB(200),TTRANS(200),TMRANS(200),
     1    TSIB(200),PSIB(200),TSIBS(200),PSIBS(200)
          COMMON SMLIM(5500),STLIM(5500),TIMEB(20,200)
          COMMON NMAT,NODES,KCYC,MAX,MID,SET,NOSPEC,NOCON,
     1    NVARH,NOSCON,NODBS,NVART,NMOGEN,NMVARG,
     1    NTOGEN,NTVARG
          COMMON SUMTIM,FOR,FORD,DELTS,DELT,DPIMAX,MAXITT,
     1    SPEED,NUTX,NUTSUM,NUTS,NUTMAX,TAU,S1,S2
          COMMON NUTAVG,PHIVRY,TEMVRY,SMALL,SMALT,BIG,KWIT,
     1    KDATA,KGOOD,MS,MSS,MSF,DELTMX,DELTO,NREG,DELTG
          COMMON RATTG,RATMG,RATT1,RATM1,RATT2, RATM2,TIMAX,DELTSS,
     1    MCYC,KSPEC,IPRINT,TIMEP
          COMMON NIT
          COMMON/SEEY/INPUT1,INPUT2,NOW
C
C
C
          IF (KCYC.GT.0) GO TO 100
          CALL REFER(NOD1,NOX1,NOCON,NODE,NODES)
          CALL REFER(NOD2,NOX2,NOCON,NODE,NODES)
          IF (KWIT.NE.0) RETURN
100       CONTINUE
          DO 200 N=1,NOCON
          N1=NOD1(N)
          N2=NOD2(N)
          J1=NODMAT(N1)
          J2=NODMAT(N2)
          IF (KODCON(N)-2)1,2,3
1         C1=KCON(J1,1)
          C2=KCON(J2,1)
          GO TO 4
2         C1=KCON(J1,2)
          C2=KCON(J2,2)
          GO TO 4
```

```
3       C1=KCON(J1,3)
        C2=KCON(J2,3)
4       CONTINUE
C       WRITE (INPUT2,*)(ATCAL(N1,LZB),LZB=1,8)
C       WRITE (INPUT2,*) N,DEL1(N),DEL2(N),C1,C2
        TRAN(N,1)=ATCAL(N1,8)*ATCAL(N2,8)*C1*C2/
     1  (ATCAL(N1,8)*C1*DEL2(N)+ATCAL(N2,8)*C2*DEL1(N))*AREA(N)
        A=ATCAL(N1,4)+ATCAL(N1,5)
        B=ATCAL(N2,4)+ATCAL(N2,5)
        TRAN(N,2)=C1*C2*A*B/(C1*A*DEL2(N)+C2*B*DEL1(N))*AREA(N)
        A=ATCAL(N1,6)+ATCAL(N1,7)
        B=ATCAL(N2,6)+ATCAL(N2,7)
        TRAN(N,3)=C1*C2*A*B/(C1*A*DEL2(N)+C2*B*DEL1(N))*AREA(N)
        TRAN(N,4)=ATCAL(N1,2)*ATCAL(N2,2)*C1*C2/(C1*ATCAL(N1,2)*DEL2(N)+
     1  ATCAL(N2,2)*DEL1(N)*C2)*AREA(N)
        A=ATCAL(N1,3)*ATCAL(N1,6)
        B=ATCAL(N2,3)*ATCAL(N2,6)
        TRAN(N,5)=-C1*C2*A*B/(C1*A*DEL2(N)+C2*B*DEL1(N))*AREA(N)
200     CONTINUE
C
C
        DO 300 N=1,NOCON
        N1=NOD1(N)
        ZIP(N1,1)=ZIP(N1,1)+TRAN(N,2)
        ZIP(N1,2)=ZIP(N1,2)+TRAN(N,3)
        ZIP(N1,3)=ZIP(N1,3)+TRAN(N,4)
        ZIP(N1,4)=ZIP(N1,4)+TRAN(N,5)
        N2=NOD2(N)
        ZIP(N2,1)=ZIP(N2,1)+TRAN(N,2)
        ZIP(N2,2)=ZIP(N2,2)+TRAN(N,3)
        ZIP(N2,3)=ZIP(N2,3)+TRAN(N,4)
        ZIP(N2,4)=ZIP(N2,4)+TRAN(N,5)
C
CAX     WRITE(INPUT2,321)
C321    FORMAT(1X,'ESOLVE ZIP(N1)')
C       WRITE(INPUT2,*)ZIP(N1,1),ZIP(N1,2),ZIP(N1,3),ZIP(N1,4)
        AFLOW=DELT*(TRAN(N,2)*(TEMP(N2)-TEMP(N1))
     1  +TRAN(N,3)*(PHI(N2)-PHI(N1))+TRAN(N,1)*(Z(N2)-Z(N1)))
        DPHI(N1)=DPHI(N1)+AFLOW/VOL(N1)
        DPHI(N2)=DPHI(N2)-AFLOW/VOL(N2)
        TFLOW=DELT*(TRAN(N,4)*(TEMP(N2)-TEMP(N1))
     1  +TRAN(N,5)*(PHI(N2)-PHI(N1)))
        DTEMP(N1)=DTEMP(N1)+TFLOW/VOL(N1)/ATCAL(N1,1)
        DTEMP(N2)=DTEMP(N2)-TFLOW/VOL(N2)/ATCAL(N2,1)
        DTF(N1)=DTF(N1)+TFLOW
        DTF(N2)=DTF(N2)-TFLOW
        DMF(N1)=DMF(N1)+AFLOW
        DMF(N2)=DMF(N2)-AFLOW
300     CONTINUE
        RETURN
        END
C
        SUBROUTINE BSCAL
C
C
C
      COMMON MATL(6),MATAB(6,10),AMX(10,8,6),
     1  AMY(10,8,6),SLOMAT(10,8,6),KCON(6,3)
      COMMON ATCAL(5500,8),NODMAT(5500),NOXMAT(5500),TEMP(5500),
     1  PHI(5500),DTEMP(5500),DPHI(5500),DTF(5500),DMF(5500)
      COMMON VOL(5500),Z(5500),NTYPE(5500),NODE(5500)
      COMMON NOTE(5500),NOXE(5500)
      COMMON NOD1(15600),NOD2(15600),NOX1(15600),NOX2(15600),
     1  DEL1(15600),DEL2(15600),AREA(15600),KODCON(15600)
      COMMON NODS(200),NODSB(200),NOXS(200),NOXSB(200),
```

```
     1  HTSURT(20,200),HMSURT(20,200),SMLOH(20,200),STLOH(20,200),
     1  HTSURE(200),HMSURE(200),LTABH(200),AREAS(200),
     1  TMVARH(20,200),TTVARH(20,200)
        COMMON PHIB(20,200),TEMPB(20,200),NODB(200),ZB(200),
     1  LTABT(200),STLOT(20,200),SMLOT(20,200)
        COMMON GMT(20,200),GM(5500),NMOXG(200),NMODG(200),
     1  TMVARG(20,200),LMTABG(200),SMLOG(20,200)
        COMMON GTT(20,200),GT(5500),NTOXG(200),NTODG(200),
     1  TTVARG(20,200),LTTABG(200),STLOG(20,200),
     1  DDPHI(5500),DDTEMP(5500),ETRORX(5500),EMRORX(5500)
        COMMON ETROR(5500),EMROR(5500)
        COMMON TRAN(15600,5),ZIP(5500,4)
        COMMON FTS(200),FMS(200),DTFS(200),DMFS(200),FTB(200),
     1  FMB(200),DMFB(200),DTFB(200),TTRANS(200),TMRANS(200),
     1  TSIB(200),PSIB(200),TSIBS(200),PSIBS(200)
        COMMON SMLIM(5500),STLIM(5500),TIMEB(20,200)
        COMMON NMAT,NODES,KCYC,MAX,MID,SET,NOSPEC,NOCON,
     1  NVARH,NOSCON,NODBS,NVART,NMOGEN,NMVARG,
     1  NTOGEN,NTVARG
        COMMON SUMTIM,FOR,FORD,DELTS,DELT,DPIMAX,MAXITT,
     1  SPEED,NUTX,NUTSUM,NUTS,NUTMAX,TAU,S1,S2
        COMMON NUTAVG,PHIVRY,TEMVRY,SMALL,SMALT,BIG,KWIT,
     1  KDATA,KGOOD,MS,MSS,MSF,DELTMX,DELTO,NREG,DELTG
        COMMON RATTG,RATMG,RATT1,RATM1,RATT2,  RATM2,TIMAX,DELTSS,
     1  MCYC,KSPEC,IPRINT,TIMEP
      COMMON NIT
        COMMON/SEEY/INPUT1,INPUT2,NOW
C
        IF (KCYC.GT.0) GO TO 100
        CALL REFER(NODS,NOXS,NOSCON,NODE,NODES)
        CALL REFER(NODSB,NOXSB,NOSCON,NODB,NODBS)
10      CONTINUE
        DO 20 N=1,NOSCON
        FTS(N)=0.0
        FMS(N)=0.0
        DTFS(N)=0.0
        DMFS(N)=0.0
        NND=NODSB(N)
        FTB(NND)=0.0
        FMB(NND)=0.0
20      CONTINUE
        DO 30 N=1,NOSCON
        N1=NODS(N)
        N2=NTYPE(N1)
        IF (N2.EQ.2) GO TO 30
        IF (N2.EQ.0) NOSPEC=NOSPEC+1
        NTYPE(N1)=1
30      CONTINUE
        DO 40 N=1,NOSCON
        HTSURE(N)=HTSURT(1,N)
        HMSURE(N)=HMSURT(1,N)
        N2=NODSB(N)
        FTB(N2)=0.0
        FMB(N)=0.0
        TTRANS(N)=AREAS(N)*HTSURE(N)
        TMRANS(N)=AREAS(N)*HMSURE(N)
40      CONTINUE
100     IF (KWIT.LT.0) GO TO 110
        DO 50 N=1,NOSCON
        N2=NODSB(N)
        DMFB(N2)=0.0
        DTFB(N2)=0.0
50      CONTINUE
        DO 60 N=1,NOSCON
        DTX=DTFS(N)
```

```
C
C
      DMX=DMFS(N)
      FTS(N)=FTS(N)+DTX
      FMS(N)=FMS(N)+DMX
      N2=NODSB(N)
      DTFB(N2)=DTFB(N2)+DTX
      DMFB(N2)=DMFB(N2)+DMX
      FTB(N2)=FTB(N2)+DTX
      FMB(N2)=FMB(N2)+DMX
60    CONTINUE
      IF(KCYC.LT.1.OR.NOW.LT.1) GO TO 110
      WRITE(INPUT2,1000)
      TX=AMAX1(SUMTIM-TAU,1.E-12)
      FTLUXS=0.0
C
      FMLUXS=0.0
      DO 70 N=1,NODBS
      FTX=FTB(N)/TX
      FMX=FMB(N)/TX
      FTLUXS=FTLUXS+FTB(N)
      FMLUXS=FMLUXS+FMB(N)
      WRITE (INPUT2,2000)NODB(N),PSIBS(N),FMB(N)
      WRITE (INPUT2,2000)NODB(N),TSIBS(N),FTB(N)
70    CONTINUE
      FTX=FTLUXS/TX
      FMX=FMLUXS/TX
      WRITE (INPUT2,6000) FTLUXS,FMLUXS
      IF(KCYC.LT.1. OR.(KCYC.NE.1.AND.KDATA.LT.1)) GO TO 110
      WRITE (INPUT2,5000)
      DO 80 N=1,NOSCON
      FTX=FTS(N)/TX
      FMX=FMS(N)/TX
      IF (MOD(N,57).EQ.1) WRITE (INPUT2,3000)
      WRITE (INPUT2,4000) NOXS(N),NOXSB(N),AREAS(N),HMSURE(N),
     1 TMRANS(N),FMS(N),FMX,DMFS(N)
      WRITE(INPUT2,4000) NOXS(N),NOXSB(N),AREAS(N),HtSURE(N),
     1 TtRANS(N),FtS(N),FtX,DtFS(N)
80    CONTINUE
      CALL LINE
110   IF (NVART.LE.0) GO TO 160
      DO 130 N=1,NODBS
      MAX=LTABT(N)
      IF (MAX.EQ.0) GO TO 130
      BTET=TSIB(N)
      BMET=PSIB(N)
      SET=SUMTIM+DELT
      IF (MAX.LT.100) GO TO 140
      PSIB(N)=SIN(6.283185*(SET+TIMEB(2,N))/TIMEB(1,N))
      PSIBS(N)=PHIB(1,N)+PHIB(2,N)*PSIB(N)
      TSIB(N)=SIN(6.283185*(SET+TIMEB(2,N))/TIMEB(1,N))
      TSIBS(N)=TEMPB(1,N)+TEMPB(2,N)*TSIB(N)
      GO TO 130
140   CALL MIDPT(TIMEB(1,N))
      TSIB(N)=TEMPB(MID,N)+STLOT(MID,N)*(SET-TIMEB(MID,N))
      PSIB(N)=PHIB(MID,N)+SMLOT(MID,N)*(SET-TIMEB(MID,N))
      TSIBS(N)=BTET+FOR*(TSIB(N)-BTET)
      PSIBS(N)=BMET+FOR*(PSIB(N)-BMET)
130   CONTINUE
160   IF (NVARH.LE.0) GO TO 220
      DO 180 N=1,NOSCON
      MAX=LTABH(N)
      IF (MAX) 190,180,200
190   MAX=-MAX
      SET=SUMTIM+FORD
```

```
      GO TO 210
200   N1=NODS(N)
      N2=NODSB(N)
      STET=0.5*(TSIBS(N2)+TEMP(N1)+DDTEMP(N1)*FORD)
      SMET=0.5*(PSIBS(N2)+PHI(N1)+DDPHI(N1)*FORD)
210   CONTINUE
      IF (MAX.GT.0) SET=STET
      CALL MIDPT(TTVARH(1,N))
      HTSURE(N)=HTSURT(MID,N)+STLOH(MID,N)*(SET-TTVARH(MID,N))
      IF (MAX.GT.0) SET=SMET
      CALL MIDPT(TMVARH(1,N))
      HMSURE(N)=HMSURT(MID,N)+SMLOH(MID,N)*(SET-TMVARH(MID,N))
180   CONTINUE
      DO 230 N=1,NOSCON
      IF (LTABH(N).EQ.0) GO TO 230
      TTRANS(N)=AREAS(N)*HTSURE(N)
      TMRANS(N)=AREAS(N)*HMSURE(N)
230   CONTINUE
220   DO 240 N=1,NOSCON
      N1=NODS(N)
      ZIP(N1,3)=ZIP(N1,3)+TTRANS(N)
      ZIP(N1,2)=ZIP(N1,2)+TMRANS(N)
      N2=NODSB(N)
      DTFS(N)=TTRANS(N)*(TSIBS(N2)-TEMP(N1))*DELT
      DMFS(N)=TMRANS(N)*(PSIBS(N2)-PHI(N1))*DELT
      DTF(N1)=DTF(N1)+DTFS(N)
      DMF(N1)=DMF(N1)+DMFS(N)
      DTEMP(N1)=DTEMP(N1)+DTFS(N)/VOL(N1)/ATCAL(N1,1)
      DPHI(N1)=DPHI(N1)+DMFS(N)/VOL(N1)
240   CONTINUE
      RETURN
3000  FORMAT(/,10X,' NODS NODSB    AREAS      HSURE      TRANS',
     1  6X,'FLOW',3X,'AVG RATE',5X,'DFS')
4000  FORMAT(10X,2I6,7E11.4)
5000  FORMAT(10X,24HEXTERNAL CONNECTION DATA)
6000  FORMAT(/10X,12HSYSTEM TOTAL,6X,4e15.5)
1000  FORMAT(10X,18HBOUNDARY NODE DATA,/,11X,4HNODB,4X,
     1  9HPOTENTIAL,4X,10HFLUID FLOW)
2000  FORMAT(10X,I5,5E15.5)
      END
C
      SUBROUTINE ISOLVE
C
C
C
      COMMON MATL(6),MATAB(6,10),AMX(10,8,6),
     1  AMY(10,8,6),SLOMAT(10,8,6),KCON(6,3)
      COMMON ATCAL(5500,8),NODMAT(5500),NOXMAT(5500),TEMP(5500),
     1  PHI(5500),DTEMP(5500),DPHI(5500),DTF(5500),DMF(5500)
      COMMON VOL(5500),Z(5500),NTYPE(5500),NODE(5500)
      COMMON NOTE(5500),NOXE(5500)
      COMMON NOD1(15600),NOD2(15600),NOX1(15600),NOX2(15600),
     1  DEL1(15600),DEL2(15600),AREA(15600),KODCON(15600)
      COMMON NODS(200),NODSB(200),NOXS(200),NOXSB(200),
     1  HTSURT(20,200),HMSURT(20,200),SMLOH(20,200),STLOH(20,200),
     1  HTSURE(200),HMSURE(200),LTABH(200),AREAS(200),
     1  TMVARH(20,200),TTVARH(20,200)
      COMMON PHIB(20,200),TEMPB(20,200),NODB(200),ZB(200),
     1  LTABT(200),STLOT(20,200),SMLOT(20,200)
      COMMON GMT(20,200),GM(5500),NMOXG(200),NMODG(200),
     1  TMVARG(20,200),LMTABG(200),SMLOG(20,200)
      COMMON GTT(20,200),GT(5500),NTOXG(200),NTODG(200),
     1  TTVARG(20,200),LTTABG(200),STLOG(20,200),
     1  DDPHI(5500),DDTEMP(5500),ETRORX(5500),EMRORX(5500)
      COMMON ETROR(5500),EMROR(5500)
```

```
      COMMON TRAN(15600,5),ZIP(5500,4)
      COMMON FTS(200),FMS(200),DTFS(200),DMFS(200),FTB(200),
     1 FMB(200),DMFB(200),DTFB(200),TTRANS(200),TMRANS(200),
     1 TSIB(200),PSIB(200),TSIBS(200),PSIBS(200)
      COMMON SMLIM(5500),STLIM(5500),TIMEB(20,200)
      COMMON NMAT,NODES,KCYC,MAX,MID,SET,NOSPEC,NOCON,
     1 NVARH,NOSCON,NODBS,NVART,NMOGEN,NMVARG,
     1 NTOGEN,NTVARG
      COMMON SUMTIM,FOR,FORD,DELTS,DELT,DPIMAX,MAXITT,
     1 SPEED,NUTX,NUTSUM,NUTS,NUTMAX,TAU,S1,S2
      COMMON NUTAVG,PHIVRY,TEMVRY,SMALL,SMALT,BIG,KWIT,
     1 KDATA,KGOOD,MS,MSS,MSF,DELTMX,DELTO,NREG,DELTG
      COMMON RATTG,RATMG,RATT1,RATM1,RATT2, RATM2,TIMAX,DELTSS,
     1 MCYC,KSPEC,IPRINT,TIMEP
      COMMON NIT
      COMMON/SEEY/INPUT1,INPUT2,NOW
C
C
      IF (NOW.LE.0) GO TO 5
      NCYC=KCYC-1
      NUTAVG=NUTSUM/MAX0(1,NCYC)
      WRITE (INPUT2,5810) NCYC,NUTSUM,NUTAVG,NUTX,FOR
5     IF (KWIT.GT.0) RETURN
      DO 10 N=1,NODES
      ETROR(N)=0.0
      EMROR(N)=0.0
      IF (NTYPE(N).EQ.0) GO TO 10
      ETRORX(N)=DELT*DDTEMP(N)
      EMRORX(N)=DELT*DDPHI(N)
10    CONTINUE
      KNOCK=0
      IF (NOCON.LT.1) GO TO 30
      DO 25 N=1,NOCON
      N1=NOD1(N)
      N2=NOD2(N)
      IF (NTYPE(N1).NE.0) GO TO 15
      IF (NTYPE(N2).EQ.0) GO TO 25
      ETROR(N2)=ETROR(N2)+DTEMP(N1)*TRAN(N,4)+
     1 DPHI(N1)*TRAN(N,5)
      EMROR(N2)=EMROR(N2)+DPHI(N1)*TRAN(N,3)+
     1 DTEMP(N1)*TRAN(N,2)
      GO TO 25
15    IF (NTYPE(N2).NE.0) GO TO 20
      ETROR(N1)=ETROR(N1)+DTEMP(N2)*TRAN(N,4)
     1 +DPHI(N2)*TRAN(N,5)
      EMROR(N1)=EMROR(N1)+DPHI(N2)*TRAN(N,3)+
     1 DTEMP(N2)*TRAN(N,2)
      GO TO 25
20    ETROR(N1)=ETROR(N1)+ETRORX(N2)*TRAN(N,4)
     1 +EMRORX(N2)*TRAN(N,5)
      ETROR(N2)=ETROR(N2)+ETRORX(N1)*TRAN(N,4)
     1 +EMRORX(N1)*TRAN(N,5)
      EMROR(N1)=EMROR(N1)+EMRORX(N2)*TRAN(N,3)
     1 +ETRORX(N2)*TRAN(N,2)
      EMROR(N2)=EMROR(N2)+EMRORX(N1)*TRAN(N,3)
     1 +ETRORX(N1)*TRAN(N,2)
      KNOCK=KNOCK+1
25    CONTINUE
30    CONTINUE
      NUTS=0
      S1=0.0
      ETSUM=0.0
      EMSUM=0.0
      HTSUM=1.E-12
      HMSUM=1.E-12
```

```
      IF (KNOCK*KCYC.GE.1) S1=SPEED
      S2=S1+1.0
      X1=S2*FORD
      DO 40 N=1,NODES
      IF (NTYPE(N).EQ.0) GO TO 40
      EX=ETROR(N)+S1*ZIP(N,3)*ETRORX(N)-ZIP(N,4)*EMRORX(N)
      BB=VOL(N)*ATCAL(N,1)
      DTEMP(N)=(BB*DTEMP(N)+FORD*EX)/(BB+FORD*S2*ZIP(N,3))
      ETRORX(N)=DTEMP(N)-ETRORX(N)
      ETSUM=ETSUM+BB*ABS(ETRORX(N))
      HTSUM=HTSUM+BB
      ETROR(N)=0.0
40    CONTINUE
      DO 50 N=1,NODES
      IF (NTYPE(N).EQ.0) GO TO 50
      EX=EMROR(N)-ZIP(N,1)*ETRORX(N)+S1*ZIP(N,2)*EMRORX(N)
      DPHI(N)=(VOL(N)*DPHI(N)+FORD*EX)/(VOL(N)+FORD*S2*ZIP(N,2))
      EMRORX(N)=DPHI(N)-EMRORX(N)
      EMSUM=EMSUM+VOL(N)*ABS(EMRORX(N))
      HMSUM=HMSUM+VOL(N)
      EMROR(N)=0.0
50    CONTINUE
      HMSUM=HMSUM*PHIVRY*5.0E-5
      HTSUM=HTSUM*TEMVRY*5.0E-5
CAX   WRITE(INPUT2,3321)
C 3321 FORMAT(1X,'ETSUM,HTSUM,EMSUM,HMSUM,KNOCK')
C     WRITE(INPUT2,*)ETSUM,HTSUM,EMSUM,HMSUM,KNOCK
      IF(ETSUM.LE.HTSUM.AND.EMSUM.LE.HMSUM.OR.KNOCK.LE.0)
     1  GO TO 4515
4400  NUTS=NUTS+1
      IF(KNOCK.LE.0) GO TO 4410
      DO 4405 N=1,NOCON
      N1=NOD1(N)
      N2=NOD2(N)
      IF(NTYPE(N1)*NTYPE(N2).EQ.0) GO TO 4405
      ETROR(N1)=ETROR(N1)+ETRORX(N2)*TRAN(N,4)
     1  +EMRORX(N2)*TRAN(N,5)
      ETROR(N2)=ETROR(N2)+ETRORX(N1)*TRAN(N,4)
     1  +EMRORX(N1)*TRAN(N,5)
4405  CONTINUE
4410  CONTINUE
4500  ETSUM=0.0
      ETMAX=0.0
      DO 4505 N=1,NODES
      IF(NTYPE(N).EQ.0) GO TO 4505
CAX   WRITE(INPUT2,3111)
C 3111 FORMAT(1X,'BB,FORD,ZIP3,ZIP4,S1,S2,ETRX,EMRX,ETRR')
      BB=VOL(N)*ATCAL(N,1)
CAX   WRITE(INPUT2,*) FORD,ZIP(N,3),ZIP(N,4),S1,S2,ETRORX(N)
C    1 ,EMRORX(N),ETROR(N)
      ETRORX(N)=(ETRORX(N)*S1*ZIP(N,3)-ZIP(N,4)*EMRORX(N)
     1  +ETROR(N))/(S2*ZIP(N,3)+BB/FORD)
      DTEMP(N)=DTEMP(N)+ETRORX(N)
      EX=ABS(ETRORX(N))
      ETSUM=ETSUM+BB*EX
      IF(NTYPE(N).NE.2) ETMAX=AMAX1(ETMAX,EX)
      ETROR(N)=0.0
4505  CONTINUE
      DO 5405 N=1,NOCON
      N1=NOD1(N)
      N2=NOD2(N)
      IF(NTYPE(N1)*NTYPE(N2).EQ.0) GO TO 5405
      EMROR(N1)=EMROR(N1)+EMRORX(N2)*TRAN(N,3)
     1  +ETRORX(N2)*TRAN(N,2)
      EMROR(N2)=EMROR(N2)+EMRORX(N1)*TRAN(N,3)
```

```
     1    +ETRORX(N1)*TRAN(N,2)
5405      CONTINUE
5410      CONTINUE
          EMSUM=0.0
          EMMAX=0.0
          DO 5505 N=1,NODES
          IF(NTYPE(N).EQ.0) GO TO 5505
          BB=VOL(N)
CAX       WRITE(INPUT2,3112)
C 3112    FORMAT(1X,'BB,FORD,ZIP2,ZIP1,S1,S2,ETRX,EMRX,EMRR')
C         WRITE(INPUT2,*) FORD,ZIP(N,2),ZIP(N,1),S1,S2,ETRORX(N)
C    1    ,EMRORX(N),EMROR(N)
          EMRORX(N)=(EMRORX(N)*S1*ZIP(N,2)-ZIP(N,1)*ETRORX(N)
     1    +EMROR(N))/(S2*ZIP(N,2)+BB/FORD)
          DPHI(N)=DPHI(N)+EMRORX(N)
          EX=ABS(EMRORX(N))
          EMSUM=EMSUM+BB*EX
          IF(NTYPE(N).NE.2) EMMAX=AMAX1(EMMAX,EX)
          EMROR(N)=0.0
5505      CONTINUE
CAX       WRITE(INPUT2,3311)
C 3311    FORMAT(1X,'ETSUM,HTSUM,EMSUM,HMSUM,ETMAX,TEMVRY,EMMAX
C    1      ,PHIVRY')
C         WRITE(INPUT2,*)ETSUM,HTSUM,EMSUM,HMSUM,ETMAX,TEMVRY,EMMAX,
C    1    PHIVRY
          IF(NUTS.LT.NUTMAX) GO TO 4510
          NUTS=0
          DPHMAX=AMAX1(DPHMAX,2*PHIVRY)
          DPHMAX=AMAX1(DPHMAX,2*TEMVRY)
          WRITE(INPUT2,5800) KCYC
          IF(DELT.GT.2*SMALL) GO TO 4515
          KWIT=10
          WRITE(INPUT2,5805)
          RETURN
4510      IF(ETSUM.GT.HTSUM.OR.EMSUM.GT.HMSUM) GO TO 4400
          IF(ETMAX.GT.5.E-4*TEMVRY.OR.EMMAX.GT.5.E-4*PHIVRY)
     1    GO TO 4400
4515      NUTSUM=NUTSUM+NUTS
          NUTX=MAX0(NUTX,NUTS)
          DPHMAX=AMAX1(DPHMAX,0.025*FLOAT(NUTS)*PHIVRY)
          DPHMAX=AMAX1(DPHMAX,0.025*FLOAT(NUTS)*TEMVRY)
          IF(NOCON.LT.1) GO TO 4620
          DO 4610 N=1,NOCON
          N2=NOD2(N)
          N1=NOD1(N)
          IF(NOX1(N).LT.0.AND.NTYPE(N1).NE.2) GO TO 4610
          IF(NTYPE(N1).NE.0) GO TO 4600
          IF(NTYPE(N2).EQ.0) GO TO 4610
          ETROR(N1)=ETROR(N1)+TRAN(N,4)*(DTEMP(N2)-DTEMP(N1))
     1    +TRAN(N,5)*(DPHI(N2)-DPHI(N1))
          EMROR(N1)=EMROR(N1)+TRAN(N,2)*(DTEMP(N2)-DTEMP(N1))
     1    +TRAN(N,3)*(DPHI(N2)-DPHI(N1))
          GO TO 4605
4600      IF(NTYPE(N2).NE.0) GO TO 4605
          ETROR(N2)=ETROR(N2)+TRAN(N,4)*(DTEMP(N1)-DTEMP(N2))
     1    +TRAN(N,5)*(DPHI(N1)-DPHI(N2))
          EMROR(N2)=EMROR(N2)+TRAN(N,2)*(DTEMP(N1)-DTEMP(N2))
     1    +TRAN(N,3)*(DPHI(N1)-DPHI(N2))
4605      FLEX=FORD*(TRAN(N,4)*(DTEMP(N2)-DTEMP(N1))
     1    +TRAN(N,5)*(DPHI(N2)-DPHI(N1)))
          DTP(N1)=DTP(N1)+FLEX
          DTP(N2)=DTP(N2)-FLEX
          FLEX=FORD*(TRAN(N,2)*(DTEMP(N2)-DTEMP(N1))
     1    +TRAN(N,3)*(DPHI(N2)-DPHI(N1)))
          DMP(N1)=DMP(N1)+FLEX
```

```
      D4F(N2)=D4F(N2)-FLEX
4610  CONTINUE
4620  DO 4630 N=1,NODES
      IF(NTYPE(N).NE.0) GO TO 4630
      DPH(N)=DPH(N)+FORD*EMROR(N)/VOL(N)
      DTEMP(N)=DTEMP(N)+FORD*ETROR(N)/VOL(N)/ATCAL(N,1)
4630  CONTINUE
      IF(NOSCON.LT.1) RETURN
      DO 4640 N=1,NOSCON
      N1=NODS(N)
      IF(NTYPE(N1).NE.2) GO TO 4640
      FLEX=FORD*TMRANS(N)*DPH(N1)
      D4F(N1)=D4F(N1)+FLEX
      D4FS(N)=D4FS(N)-FLEX
      FLEX=FORD*TTRANS(N)*DTEMP(N1)
      DTF(N1)=DTF(N1)+FLEX
      DTFS(N)=DTFS(N)-FLEX
4640  CONTINUE
      RETURN
5800  FORMAT(/10X,6HKCYC = ,I3,5X,'  MASS BALANCE ITERATION'1X,
     1  'FAILED TO CONVERGE. THIS MAY BE CAUSED BY A CONNECTED'1X
     1  ,'SERIES',/,10X,'OF NODES WITH TIME CONSTANCE MUCH SMALLER',
     1  1X,'THAN DELT.')
5805  FORMAT(/,10X,'CAN NOT REDUCE DELT, WILL END PROBLEM')
5810  FORMAT(/,10X,'KCYC= ',I4,'  ITERATIONS...  TOTAL=',
     1  I5,'  AVERAGE=',I3,'  MAXIMUM= ',I3,'   FOR=',F6.3)
      END
C
      SUBROUTINE TIME
C
C
C
      COMMON MATL(6),MATAB(6,10),AMX(10,8,6),
     1  AMY(10,8,6),SLOMAT(10,8,6),KCON(6,3)
      COMMON ATCAL(5500,8),NODMAT(5500),NOXMAT(5500),TEMP(5500),
     1  PH(5500),DTEMP(5500),DPH(5500),DTF(5500),D4F(5500)
      COMMON VOL(5500),Z(5500),NTYPE(5500),NODE(5500)
      COMMON NOTE(5500),NOXE(5500)
      COMMON NOD1(15600),NOD2(15600),NOX1(15600),NOX2(15600),
     1  DEL1(15600),DEL2(15600),AREA(15600),KODCON(15600)
      COMMON NODS(200),NOL(SB(200),NOXS(200),NOXSB(200),
     1  HTSURT(20,200),HMSURT(20,200),SMLO4(20,200),STLO4(20,200),
     1  HTSURE(200),HMSURE(200),LTABH(200),AREAS(200),
     1  TMVARH(20,200),TTVARH(20,200)
      COMMON PHIB(20,200),TEMPB(20,200),NODB(200),ZB(200),
     1  LTABT(200),STLOT(20,200),SMLOT(20,200)
      COMMON GMT(20,200),GM(5500),NMOXG(200),NMODG(200),
     1  TMVARG(20,200),LMTABG(200),SMLOG(20,200)
      COMMON GTT(20,200),GT(5500),NTOXG(200),NTODG(200),
     1  TTVARG(20,200),LTTABG(200),STLOG(20,200),
     1  DDPH(5500),DDTEMP(5500),ETRORX(5500),EMRORX(5500)
      COMMON ETROR(15600),EMROR(5500)
      COMMON TRAN(15600,5),ZIP(5500,4)
      COMMON FTS(200),FMS(200),DTFS(200),D4FS(200),FTB(200),
     1  FMB(200),D4FB(200),DTFB(200),TTRANS(200),TMRANS(200),
     1  TSIB(200),PSIB(200),TSIBS(200),PSIBS(200)
      COMMON SMLD4(5500),STLD4(5500),TIMEB(20,200)
      COMMON NMAT,NODES,KCYC,MAXX,MID,SET,NOSPEC,NOCON,
     1  NVARH,NOSCON,NODBS,NVART,NMOXGEN,NMVARG,
     1  NTOGEN,NTVARG
      COMMON SUMTD4,P2R,FORD,DELTS,DELT,DPHMAX,MAXITT,
     1  SPEED,NUTX,NUTSUM,NUTS,NUTMAX,TAU,S1,S2
      COMMON NUTAVG,PHIVRY,TEMVRY,SMALL,SMALT,BIG,KWIT,
     1  KDATA,KGOOD,MS,MSS,MSP,DELTMX,DELTO,NREG,DELTG
      COMMON RATTG,RATM0,RATT1,RATM1,RATT2, RATM2,TIMAX,DELTSS,
```

```
     1  MCYC,KSPEC,IPRINT,TIMEP
        COMMON /IT/
        COMMON/SEEY/INPUT1,INPUT2,NOW
C
C
          IF(KCYC.GE.1) GO TO 4200
          IF(KWTT.GT.0) RETURN
          KGOOD=0
          MF=0
          MS=0
          IX=0
          DO 39 I=1,8
          DO 49 J=1,NMAT
          IX=IX+MATAB(I,J)
49        CONTINUE
39        CONTINUE
          NCHANG=NVART+NVARH+NTVARG+NMVARG+IX
          MSS=0
          NPRINT=0
          DELTO=0.0
          DPDMAX=0.0
          SMALT=SMALL
          RATTO=1.0
          RATMO=1.0
          IF(KCYC.LE.1) DPDMAX=0.0
          DELTMX=AMAX1(1.E-10,DELTO)
          BIG=0.9999E12
          GO TO 4210
4200      IF(KCYC.LE.1) DPDMAX=0.0
          IF(KWTT.GT.0) GO TO 4710
          NOW=0
          KWTT=0
          IF(DELTMX.GE.BIG.OR.NCHANG*(KCYC-1).GT.0.AND.KSPEC.LT.0)
       1  GO TO 4235
4210      DELTMX=1.5E12
          NREC=0
          DO 4220 N=1,NODES
          SMLIM(N)=1.E24
          STLIM(N)=1.E24
          IF(ZIP(N,2).GT.0.0) SMLIM(N)=AMAX1(1.E-24,VOL(N)/ZIP(N,2))
          IF(ZIP(N,3).GT.0.0)STLIM(N)=AMAX1(1.E-24,(VOL(N)*ATCAL(N,1)
       1  /ZIP(N,3)))
          IF(NTYPE(N).NE.0) GO TO 4220
          TSLIM=AMIN1(SMLIM(N),STLIM(N))
CAX       WRITE(INPUT2,6969)
C 6969    FORMAT(1X,'TSLIM')
C         WRITE(INPUT2,*)TSLIM,SMLIM(N),STLIM(N)
          NREC=NREC+1
          DELTMX=AMIN1(DELTMX,TSLIM)
4220      CONTINUE
          DELTMX=AMAX1(DELTMX*0.66666,1.E-10)
CAX       WRITE(INPUT2,6968)
C 6968    FORMAT(1X,'DELT AFTER TSLIM')
C         WRITE(INPUT2,*)DELT
          IF(DELTMX.LT.DELTO) GO TO 4225
          DELTMX=DELTO
          GO TO 4235
4225      IF(MS*NREC.EQ.0.OR.KSPEC.LT.0) GO TO 4235
          NEWS=0
          X1=1.8*DELTMX
          DO 4230 N=1,NODES
          TSLIM=AMIN1(SMLIM(N),STLIM(N))
          IF(NTYPE(N).NE.0.OR.X1.LT.TSLIM)
       1  GO TO 4230
          IF(NODE(N).LT.0) GO TO 4230
```

```
      NTYPE(N)=4
      NOSPEC=NOSPEC+1
      NEWS=1
c     WRITE(INPUT2,5800) KCYC,NODE(N)
4230  CONTINUE
      IF(NEWS.EQ.0) GO TO 4235
      KGOOD=0
      MS=0
      GO TO 4210
4235  SMALL=AMAX1(SMALL,SMALT)
      IF(SMALL.GE.DELTMX) SMALL=DELTMX*0.9999
      IF(BIG*SMALT.LE.1.AND.NODES.LE.4*NREG.AND.DELTMX.LE.BIG)
     1 SMALL=DELTMX*0.01
4300  DPHIS=1.0E-24
      DTEMS=1.0E-24
      DO 4310 N=1,NODES
      IF(NTYPE(N).EQ.2.OR.NODE(N).LE.0.OR.KCYC.LE.1) GO TO 4310
      DPHIS=AMAX1(DPHIS,ABS(DPHI(N)))
      DTEMS=AMAX1(DTEMS,ABS(DTEMP(N)))
4310  CONTINUE
      IF(DELTG.EQ.0.0) DELTG=1.E-15
      DELTSS=DELTG
      DELTS=DELT
      RATT2=RATTG
      RATM2=RATMG
      RATT1=PHIVRY/DPHIS
      RATM1=TEMVRY/DTEMS
      TRY11=TEMVRY/AMAX1(DPIMAX,DTEMS)
      TRY12=PHIVRY/AMAX1(DPIMAX,DPHIS)
      RATIO=AMIN1(TRY11,TRY12)
      RATIOS=RATIO
CAX   WRITE(INPUT2,6967)
C 6967 FORMAT(1X,'RATIO DELT DTEMS DPHIS')
C     WRITE(INPUT2,*)RATIO,DELT,DTEMS,DPHIS
      IF(RATIO.GT.0.501.OR.DELT.LT.1.01*SMALL) GO TO 4315
      WRITE(INPUT2,5990) KCYC,DPIMAX,DTEMS,DPHIS,DELT,SUMTIM
      KWIT=-1
      KGOOD=0
      FOR=1.0
      GO TO 4320
4315  KGOOD=KGOOD+1
CAX   WRITE(INPUT2,6966)
C 6966 FORMAT(1X,'KGOOD')
C     WRITE(INPUT2,*)KGOOD
      SUMTIM=SUMTIM+DELT
      DELTG=DELTS
      RATTG=RATT1
      RATMG=RATM1
      DPMAXS=DPIMAX
      DMMAXS=DPIMAX
4320  DPIMAX=0.0
      IF(KCYC.GT.0) GO TO 4325
      IF(NREG.LE.0) RATIO=RATIO*0.01
      GO TO 4335
4325  IF(RATIO.GT.1) GO TO 4330
      RATIO=AMAX1(RATIO*RATIO,0.5)
      GO TO 4335
4330  RATIO=AMIN1((RATIO+1.)*0.5,2.0)
4335  DELT=DELTS*RATIO
CAX   WRITE(INPUT2,6965)
C6965 FORMAT(1X,'RATIO DELT')
C     WRITE(INPUT2,*) RATIO,DELTS
      IF(DELT.GE.SMALL) GO TO 4340
      DELT=SMALL
      MF=MF+1
```

```
      GO TO 4400
4340  IF(DELT.LE.DELTMX) GO TO 4400
      DELT=DELTMX
      MSS=MSS+1
      MS=MS+1
4400  FORD=FOR*DELT
      IF(KWIT.LT.0) RETURN
      IF(TIMEP.LE.0.0) GO TO 4425
      TIMS=TIMEP*AMOD(SUMTIM,TIMEP)
      TIM=TIMS+TIMEP*FLOAT(INT(1.+(0.666*DELT-TIMS)/TIMEP))
      PIX=FLOAT(INT(TIM/DELT+0.5))
      DELT=TIM/PIX
      IF(DELT.LT.SMALL) DELT=TIM/AMAX1(PIX-1.,1.)
      IF(DELT.GT.DELTMX) DELT=TIM/(PIX+1.)
      DELT=AMAX1(DELT,SMALL)
4425  IF(TIMAX.GT.0..AND.TIMAX-SUMTIM.LT.DELT)
     1  DELT=TIMAX-SUMTIM+1.E-12
      RASTT=1.0
      RASTM=1.0
      IF(KGOOD.LE.1) GO TO 4440
      SLOPS=RATT1*DELTS/(RATT2*DELTSS)
      RDELT=(DELT+DELTS)/(DELTS+DELTSS)
      IF(SLOPS-1.) 4430,4440,4435
4430  RASTT=1.+RDELT*(1.-SLOPS)
      GO TO 4440
4435  RASTT=SLOPS**(-RDELT)+1.0E-24
4440  FOR1=AMAX1(0.57,AMAX1(1.0,RASTT)/(1.0+RASTT))
      IF(KGOOD.LE.1) GO TO 9440
      SLOPS=RATM1*DELTS/(RATM2*DELTSS)
CAX   WRITE(INPUT2,6964)
C 6964 FORMAT(1X,'RATM2,RATT2,DELTSS')
C     WRITE(INPUT2,*)RATM2,RATT2,DELTSS
      IF(SLOPS-1.)9430,9440,9435
9430  RASTM=1.+RDELT*(1.-SLOPS)
      GO TO 9440
9435  RASTM=SLOPS**(-RDELT)+1.0E-24
9440  FOR2=AMAX1(0.57,AMAX1(1.0,RASTM)/(1.0+RASTM))
      FOR=AMIN1(FOR2,FOR1)
CAX   WRITE(INPUT2,6963)
C 6963 FORMAT(1X,'FOR FOR1 FOR2')
C     WRITE(INPUT2,*)FOR,FOR1,FOR2

      IF(KSPEC-2) 4500,4442,4445
4442  FOR=1.0
      GO TO 4500
4445  FOR=0.5
4500  PHMIN1=1.0E10
      PHMAX1=-1.0E10
      PTMIN1=1.0E10
      PTMAX1=-1.0E10
      FORD=FOR*DELT
      DO 4515 N=1,NODES
      PHI(N)=PHI(N)+DPHI(N)
      TEMP(N)=TEMP(N)+DTEMP(N)
      PHMAX1=AMAX1(PHMAX1,PHI(N))
      PHMIN1=AMIN1(PHMIN1,PHI(N))
      PTMAX1=AMAX1(PTMAX1,TEMP(N))
      PTMIN1=AMIN1(PTMIN1,TEMP(N))
      DEXT=RASTT*DTEMP(N)/DELTS
      DEXM=RASTM*DPHI(N)/DELTS
      IF(NTYPE(N).EQ.2.OR.NODE(N).LT.0) GO TO 4515
      IF(NTYPE(N).EQ.0.OR.KCYC.GT.1) GO TO 4505
      DEXT=DEXT/(DELTS/STLIM(N)+1.)
      DEXM=DEXM/(DELTS/SMLIM(N)+1.)
4505  IF(DEXT**DDTEMP(N).LT.0.0) DEXT=1.0E-10*DEXT
```

```
      IF(DEXM**DDPHI(N).LT.0.0) DEXM=1.0E-10*DEXM
      DDPHI(N)=DEXM
      DDTEMP(N)=DEXT
CAX   WRITE(INPUT2,6961)
C 6961 FORMAT(1X,'DDPHI,DDTEMP')
C     WRITE(INPUT2,*)DEXM,DEXT
4515  CONTINUE
      IF(TIMAX.EQ.0.OR.TIMAX.GT.SUMTIM.AND.SUMTIM.GE.0.)
     1 GO TO 4610
      KWTT=1
      GO TO 4710
4610  IF(KSPEC.LT.0.OR.(DELTMX.LT.AMIN1(BIG,DELTO).AND.
     1 DELTMX.GT.DELT).OR.RATIOS.LT.1.E3.OR.KCYC.LT.10.
     1 OR.KGOOD.LT.3) GO TO 4620
      KWTT=4
      GO TO 4710
4620  IF(MCYC.EQ.0.OR.MCYC.GT.KCYC.AND.KCYC.GE.0) GO
     1 TO 4700
      KWTT=7
      GO TO 4710
C 4640 IF(MSEC) 4645,4700,4642
C     ==========TIMER======
C     IF(KSECS.LT.MSEC) GO TO 4700
C 4645 KWTT=8
C     GO TO 4710
 4700 IF(KWTT)4810,4702,4710
 4702 IF(XCYC.LE.1) GO TO 4710
      IF(TIMEP.LE.0.) GO TO 4705
      IF(0.666667*DELT.LT.AMIN1(TIMEP-TIMS,TIMS)) GO
     1 TO 4705
      NOW=2
      GO TO 4712
4705  IF(IPRINT.EQ.0) GO TO 4810
      IF(MOD(KCYC,IPRINT).GT.0) GO TO 4810
4710  NOW=1
4712  NPRINT=NPRINT+1
      CALL LINE
      WRITE(INPUT2,5825)NPRINT,KCYC,MF,MSS,KWTT,DELTMX,SMALL,
     1 PHIVRY,DPHIS,TEMVRY,DTEMS,NUTS
      WRITE(INPUT2,5555)
      WRITE(INPUT2,5830) SUMTIM,DELTS
      CALL LINE
      WRITE(INPUT2,5895)
      WRITE(INPUT2,5896) (NODE(N),PHI(N),N=1,NODES)
5896  format(1x,1817.2)
4810  RETURN
5800  FORMAT(/10X,5HCYCLE,I5,10H MADE NODE,I4,15H A SPECIAL NODE)
5825  FORMAT(/10X,'PRINTOUT CYCLE  MF  MSS  KWTT  DELTMX'
     1 ,7X,'SMALL',8X,'PHIVRY',8X,'DMPHI',8X,'TEMVRY',7X,'DMTEM',
     1 7X,'NUTS'
     1 /,10X,516,6E13.5,I5)
5830  FORMAT(8X,2E15.5/,8X,6E15.5)
5895  FORMAT(/,10X,'NODE',10X,'PHI',13X,'NODE',10X,'PHI',
     1 13X,'NODE',10X,'PHI',13X,'NODE',10X,'PHI')
5990  FORMAT(10X,'WILL REPEAT CYCLE',I3,'  DPIMAX=',F10.5,' DTEMS='
     1 ,F10.6,' DPHIS=',F10.5,' DELT=',F10.5,'   SUMTIM=',F10.5)
c5896 format(1x,1817)4,6X,E12.5))
5899  format(1x,1817)
XXXXX       c5896       format(/,8(i5,' ',e10.3))
6666  FORMAT(/,10X,'NODE',9X,'TEMP',13X,'NODE',9X,'TEMP',
     1 13X,'NODE',9X,'TEMP',13X,'NODE',9X,'TEMP')
5555  FORMAT(/16X,'TIME',11X,'DELTIME')
      END
C
      SUBROUTINE REFER(LIST,LISTX,MAXE,LISTR,MAR)
```

```
C
CC
CCC
CCCC
CCCCC
CCCCCC
C
      DIMENSION LIST(15600),LISTX(15600),LISTR(15600)
C
C
      DO 1 N=1,MAXE
      J=MAR
      DO 2 L=1,MAR
      IF(LISTX(N).EQ.LISTR(J)) GO TO 3
2     J=J-1
3     LIST(N)=J
1     CONTINUE
      RETURN
      END
c
C
      SUBROUTINE MIDPT(VECT)
C
CC
CCC
      COMMON MATL(6),MATAB(6,10),AMX(10,8,6),
     1 AMY(10,8,6),SLOMAT(10,8,6),KCON(6,3)
      COMMON ATCAL(5500,8),NODMAT(5500),NOXMAT(5500),TEMP(5500),
     1 PHI(5500),DTEMP(5500),DPHI(5500),DTF(5500),DMF(5500)
      COMMON VOL(5500),Z(5500),NTYPE(5500),NODE(5500)
      COMMON NOTE(5500),NOXE(5500)
      COMMON NOD1(15600),NOD2(15600),NOX1(15600),NOX2(15600),
     1 DEL1(15600),DEL2(15600),AREA(15600),KODCON(15600)
      COMMON NODS(200),NODSB(200),NOXS(200),NOXSB(200),
     1 HTSURT(20,200),HMSURT(20,200),SMLOH(20,200),STLOH(20,200),
     1 HTSURE(200),HMSURE(200),LTABH(200),AREAS(200),
     1 TMVARH(20,200),TTVARH(20,200)
      COMMON PHIB(20,200),TEMPB(20,200),NODB(200),ZB(200),
     1 LTABT(200),STLOT(20,200),SMLOT(20,200)
      COMMON GMT(20,200),GM(5500),NMOXG(200),NMODG(200),
     1 TMVARG(20,200),LMTABG(200),SMLOG(20,200)
      COMMON GTT(20,200),GT(5500),NTOXG(200),NTODG(200),
     1 TTVARG(20,200),LTTABG(200),STLOG(20,200),
     1 DDPHI(5500),DDTEMP(5500),ETRORX(5500),EMRORX(5500)
      COMMON ETROR(5500),EMROR(5500)
      COMMON TRAN(15600,5),ZIP(5500,4)
      COMMON FTS(200),FMS(200),DTFS(200),DMFS(200),FTB(200),
     1 FMB(200),DMFB(200),DTFB(200),TTRANS(200),TMRANS(200),
     1 TSIB(200),PSIB(200),TSIBS(200),PSIBS(200)
      COMMON SMLIM(5500),STLIM(5500),TIMEB(20,200)
      COMMON NMAT,NODES,KCYC,MAX,MID,SET,NOSPEC,NOCON,
     1 NVARH,NOSCON,NODBS,NVART,NMOGEN,NMVARG,
     1 NTOGEN,NTVARG
      COMMON SUMTIM,FOR,FORD,DELTS,DELT,DPIMAX,MAXITT,
     1 SPEED,NUTX,NUTSUM,NUTS,NUTMAX,TAU,S1,S2
      COMMON NUTAVG,PHIVRY,TEMVRY,SMALL,SMALT,BIG,KWIT,
     1 KDATA,KGOOD,MS,MSS,MSF,DELTMX,DELTO,NREG,DELTG
      COMMON RATTG,RATMG,RATT1,RATM1,RATT2, RATM2,TIMAX,DELTSS,
     1 MCYC,KSPEC,IPRINT,TIMEP
     COMMON NIT
      COMMON/SEEY/INPUT1,INPUT2,NOW
C
C
      DIMENSION VECT(1)
      MIN=1
```

```
1       MID=(MIN+MAX)/2
        IF(SET-VECT(MID)) 2,4,3
2       MAX=MID
        IF(MAX-2) 4,1,1
3       MIN=MID
        IF(MAX-MIN.GE.2) GO TO 1
4       RETURN
        END
```

APPENDIX B

MATERIAL PROPERTIES

With the application of a vertical hydraulic load to a radially confined soil element, the stress state within the soil element changes. This change in stress state due to the change in pore-air and pore-water pressures within the soil element is shown in Figure B-1.

In order to predict the immediate deformation and the subsequent time-dependent deformation of a clay-based backfill subjected to a hydraulic load, the material properties of the backfill along the stress-state paths AB, BE, and EF, shown in Figure B-1, have to be defined. In this section, the laboratory methods used to determine these parameters and the results are presented.

Stress-State Path AB

Along stress-state path AB deformation of the soil responds immediately to the applied load. In this study, the stress-state path AB was determined by performing compression tests on the backfill specimen in a 5-cm diameter oedometer ring under different applied loads. The corresponding suction within the soil under the different applied loads were measured with a psychrometer. The resulting stress-state path, AB, projected on the strain versus $(\sigma - u_a)$, and the $(\sigma - u_a)$ versus $(u_a - u_w)$ planes are plotted in Figures B-2 and B-3, respectively.

Stress-State Path EF

The stress-state path EF was determined by performing oedometer tests on the backfill specimen with initial moisture content at its liquid limit. The standard oedometer ring was substituted with a 5-cm long stainless-steel ring in order to accommodate the anticipated large vertical deformation of the backfill. A summary of the test results is presented in Table B-1 and the resulting stress-state path is plotted in Figure B-4.

Stress-state Path BE

With the determination of the stress-state paths AB and EF, the initial and final stress-states of the stress-state path BE are defined. In this study, it is assumed that the stress-state path between the stress-state points B and E is a straight line. Thus the compressibility, m_2^s, of the backfill along the stress-state path BE is equal to the slope of the line BE. And the compressibility of the pore-air, $m_2^{a_r}$ is equal to the ratio of the volume of the pore-air/volume of the soil element to the total suction within the soil element.

TABLE B-1 SUMMARY OF THE TEST DATA

LOAD (MPa)	HEIGHT OF SPECIMEN (cm)	t_{90} (min)	VERTICAL STRAIN * %	MOISTURE CONTENT %	c_v (m^2/yr)
0.008	1.71	49	32.0	35.5	0.64
0.10	1.66	28	16.1	26.45	0.95
0.19	1.46	26	11.4	23.78	0.86
0.38	1.40	27	6.7	21.11	0.76
0.77	1.34	27	1.6	18.21	0.70
1.53	1.28	25	-3.2	15.48	0.68
3.07	1.22	26	-8.0	12.75	0.59
	1.16				

* assumed zero strain at reference density

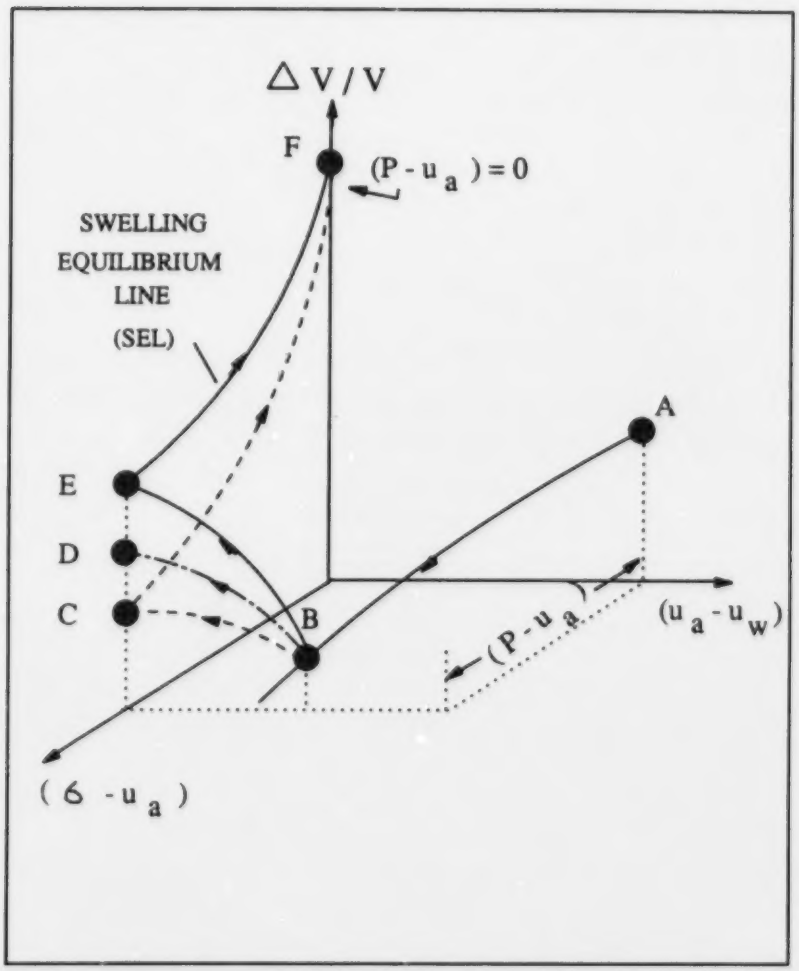

FIGURE B-1 STRESS-STATE PATH OF PARTIALLY SATURATED SWELLING SOIL.

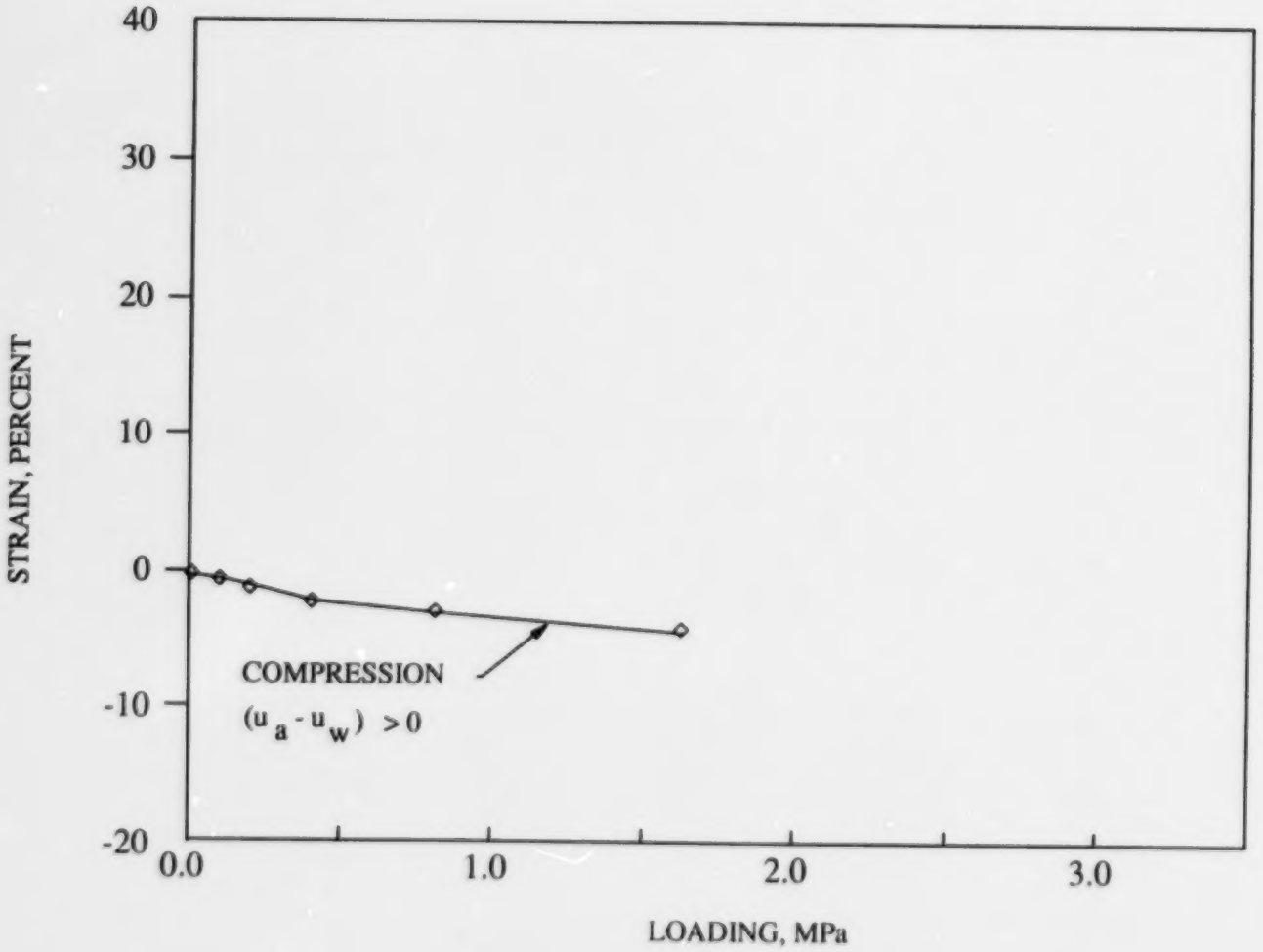

FIGURE B-2 THE COMPRESSION CURVE OF THE CLAY-BASED BACKFILL - PROJECTION ON THE STRAIN versus $(\sigma - u_a)$ PLANE.

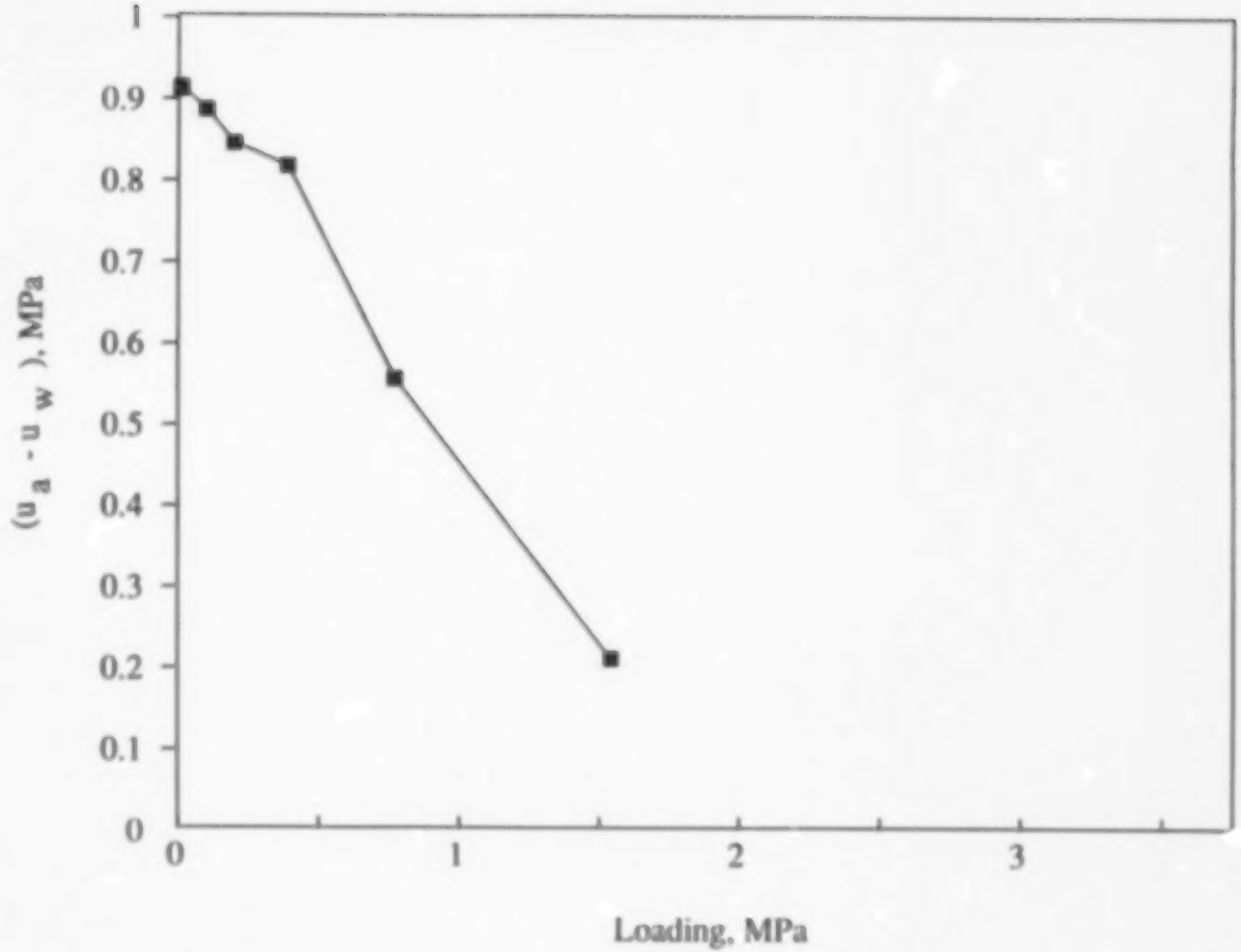

FIGURE B-3 THE COMPRESSION CURVE OF THE CLAY-BASED BACKFILL - PROJECTION ON THE ($\sigma - u_a$) versus ($u_a - u_w$) PLANE.

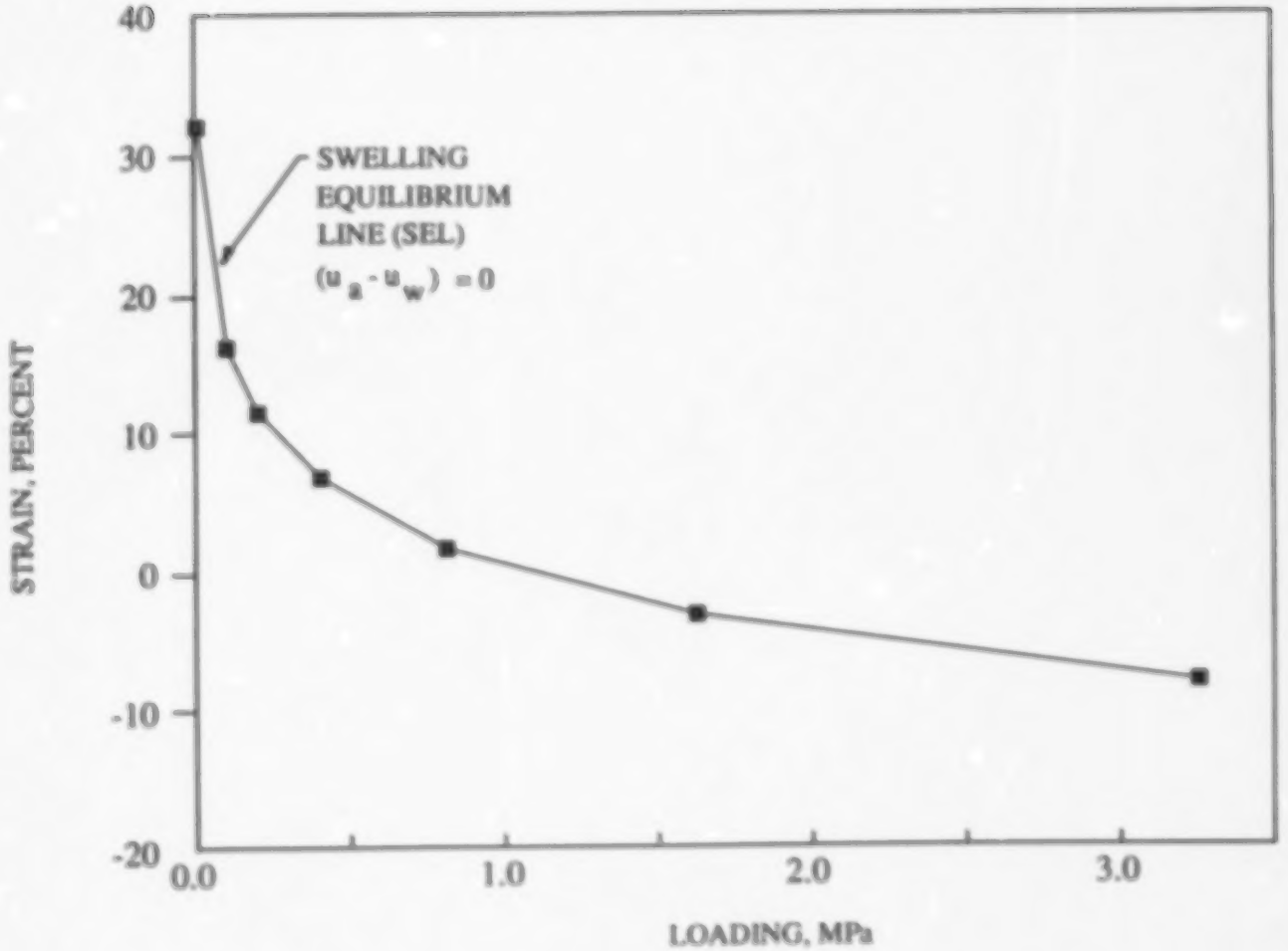

FIGURE B-4 THE SWELLING EQUILIBRIUM LINE (SEL) OF THE BACKFILL.